UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Smart E-Tickets: Buying Authentic and Trustworthy Tickets with Blockchain

Daniel Filipe Martiniano dos Santos

Mestrado em Engenharia Informática
Especialização em Engenharia de Software

Dissertação orientada por:
Prof. Doutora Maria Antónia Bacelar da Costa Lopes

2019

Acknowledgements

Agradeço aos meus pais e à minha irmã por terem tido a paciência e de me terem dado o suporte depois de ter inicialmente errado a nível académico na escolha do caminho a tomar e pelas condições que me deram para te sucesso nesta etapa da minha vida.

Agradeço aos meus amigos de sempre, por todos os momentos e por todo o suporte que sempre me deram ao longo da licenciatura e mestrado. Um grande agradecimento especialmente ao Marcus Dias e Miguel Espírito Santo, por todo o apoio que me deram neste mestrado.

Agradeço à minha orientadora, a Professora Doutora Antónia Bacelar da Costa Lopes, por me ter ajudado ao longo deste mestrado e dos conselhos dados tanto a nível académico como para o meu futuro a nível profissional.

Agradeço à Accenture pela oportunidade que me deram para explorar uma nova tecnologia e a sua aplicabilidade. Agradeço aos meus supervisores, Tiago Minchin e especialmente ao Álvaro Silva, por todo o suporte que me deram durante o tempo todo que passei na empresa. Agradeço aos meus colegas que realizaram as suas teses de mestrado na empresa, pelo bom ambiente e pelas horas de almoço cheias de boa animação.

Aos meus amigos e família.

Abstract

The price of tickets in the secondary market is a problem for event organizers and authorized resellers because it allows for ordinary people to make an illegal profit out of these tickets. Moreover, it can reduce the trust of consumers when they are trying to buy tickets. A technological solution that prevents illegal ticket sales and ensures that consumers can trust the ticket they have just bought and can resell it legally if they cannot attend the show, would be beneficial to consumers, event organizers, authorized resellers and the government. An online marketplace built around digital tickets linked to their owners, which cannot be lost and have a proof of ownership, will improve consumers' confidence when buying a ticket. Blockchain, a type of distributed ledger, has open new opportunities to change and enhance trade-related problems. Making ticket sales 100% safe, fraud free, no duplication of tickets and the decrease of secondary market business and their exorbitant prices, may solve a recurrent problem that affects different businesses and consumers. The main objective of this project is to develop a web platform that will use blockchain technology to allow for safer ticket sales, create a marketplace for authorized resellers and event organizers, and to have a legal process of reselling tickets.

Keywords: Blockchain Technology, Smart E-Tickets, Cryptocurrency, Hyperledger Fabric, Trust

Resumo Adicional

O preço dos bilhetes vendidos no mercado paralelo é um problema para os organizadores de eventos e para os pontos de venda autorizados porque permite que qualquer pessoa possa lucrar com a venda ilegal de bilhetes. Este problema tende a reduzir a confiança dos consumidores na compra de bilhetes. O crescente número de eventos realizados anualmente exacerba ainda mais o problema: torna-se mais difícil ter confiança nos bilhetes comprados e na capacidade, caso se torne necessário, de os conseguir revender legalmente. Permitir que os bilhetes sejam comprados e revendidos através de uma plataforma *web*, que assegure a autenticidade e a confiança nos bilhetes, pode aumentar a confiança dos consumidores quando estes pretendem comprar bilhetes para um evento específico. A utilização de *blockchain*, um tipo de *distributed ledger*, abriu novas oportunidades para alterar e melhorar os problemas relacionados com o comércio. O objetivo principal é desenvolver uma plataforma *web* que recorre à tecnologia *blockchain* para permitir vendas de bilhetes mais seguras, e que oferece um mercado onde vendedores autorizados e organizadores de eventos podem vender os seus bilhetes e que assegura que a revenda de bilhetes ocorre de acordo com a lei em vigor.

Várias empresas interessadas na indústria de venda de bilhetes têm analisado os processos atuais de venda e revenda de bilhetes e procurado soluções que permitam que os consumidores possam comprar bilhetes da forma mais cómoda e segura possível. Em particular existem já alguns projetos nesta área que exploraram se, e como, o uso da tecnologia de *blockchain* pode trazer vantagens para todo o processo de venda e revenda de bilhetes. Um exemplo disto é o *GUTS Ticket*, um sistema de venda e revenda de bilhetes que utiliza a plataforma pública de *blockchain Ethereum*, e que tira partido de uma criptomoeda para as transações que envolvem os bilhetes. A utilização de uma criptomoeda permite endereçar o problema dos custos associados ao sistema e à infraestrutura de que este necessita para funcionar. Por outro lado, ao usarem uma criptomoeda, estes sistemas sujeitos às oscilações do mercado. Tendo em conta as necessidades e os requisitos propostos para este projeto, decidiu-se explorar a utilização de uma plataforma de *blockchain* privada — a plataforma da IBM, Hyperledger Fabric— e analisar se, e como, a utilização este tipo de plataforma privada poderá ser vantajosa em relação aos projetos que existem atualmente.

Este projeto foi desenvolvido sob a alçada da Accenture e tendo como objetivo adquirir conhecimento sobre a tecnologia de *blockchain* e desenvolver uma prova de conceito sobre um caso de uso a definir. Como tal, a equipa composta por três elementos, todos a realizar Projeto em Engenharia Informática, começou por investigar e analisar diferentes projetos que estavam a ser realizados dentro e fora da empresa recorrendo à tecnologia de *blockchain*. Após esta investigação e com o conhecimento adquirido, a equipa identificou um conjunto de possíveis casos de uso que poderiam vir a beneficiar do uso de *blockchain*. Os casos de uso identificados, cerca de cinquenta, são de indústrias tão diversas como indústria das comunicações, media e tecnologia, indústria da saúde, indústria de produtos, serviços públicos, indústria agrícola e indústria de serviços financeiros. Os casos de uso identificados foram posteriormente sujeitos a um processo de avaliação tendo em consideração diferentes dimensões, nomeadamente o interesse no projeto, a dificuldade de implementação, a relevância para a empresa, a relevância do papel da *blockchain*, os aspetos legais envolvidos, número de intervenientes, a interoperabilidade com os sistemas atuais, a abertura dos parceiros e negócios a adotar um novo sistema e a exposição de mercado. Esta avaliação guiou a selecção de três casos de uso que os três elementos da equipa iriam explorar no seu trabalho de mestrado. Um dos casos de uso escolhidos foi a venda e revenda de bilhetes, o qual é endereçado neste trabalho. Os outros dois casos de isso são em duas áreas diferentes; um é na área da saúde e o outro na área de serviços financeiros.

A solução proposta neste trabalho passa pela criação de uma plataforma web que funcione como um mercado online de venda e revenda de bilhetes, onde organizadores de eventos, revendedores de bilhetes licenciados e os seus clientes, possam vender e revender bilhetes tirando partido das características da tecnologia de *blockchain*. A tecnologia de *blockchain* é conhecida por eliminar os intermediários dos processos atuais. No entanto, considerou-se que não era vantajoso retirar do processo de venda de bilhetes os revendedores licenciados, visto que estes têm um público alvo bem definido, têm processos de marketing e publicidade bem preparados e, assim sendo, podem trazer mais utilizadores para a aplicação e ser um fator importante para o seu sucesso. Um organizador de eventos pode através desta aplicação vender bilhetes para os seus eventos ou utilizar a plataforma para distribuir os bilhetes pelos revendedores de bilhetes licenciados que estejam interessados em vender bilhetes para os seus eventos. A plataforma tem como um dos seus grandes objetivos combater a fraude que acontece atualmente na revenda de bilhetes por pessoas não autorizadas, e que envolve tanto a venda de bilhetes a preços exorbitantes como a venda de bilhetes falsos. Este problema é combatido através da fixação dos bilhetes à plataforma: os bilhetes transacionados apenas existem na plataforma e a única maneira de os vender, e revender, é através da plataforma. Quando um bilhete é emitido pelo organizador do evento, este define o preço final do bilhete e este é o preço pelo qual ele pode ser revendido mais tarde por um revendedor de bilhetes licenciado ou por um cliente. Assim, um cliente que compre um bilhete para um evento, e que por alguma razão não possa ir ao evento, pode colocar o seu bilhete para venda na plataforma. O preço com que este bilhete é colocado à venda é o mesmo que foi definido quando o organizador do evento emitiu o bilhete na plataforma e não pode ser alterado. A plataforma oferece ainda a opção de venda de bilhetes em lote de modo a que os organizadores de eventos possam colocar à venda bilhetes exclusivamente para revendedores licenciados.

Para um cliente poder entrar no recinto do evento para o qual comprou um bilhete tem de mostrar, ao responsável pela verificação dos bilhetes, o seu bilhete que está disponível na plataforma. Porque as transações que envolvem a *blockchain* são demoradas e há limitações sérias no número de transações por segundo que a *blockchain* consegue suportar, para a validação dos bilhetes foi considerada a possibilidade de serem descarregados os bilhetes para uma base de dados auxiliar. Isto levaria a que fosse necessário fechar a venda e revenda dos bilhetes de um evento horas antes do evento começar, altura em que os bilhetes seriam não só descarregados para a base de dados auxiliar como passavam para o estado de resgatados. Esta solução, apesar de trazer inconvenientes para clientes e revendedores, permite evitar o fluxo de pedidos de validação de bilhetes que a *blockchain* sofreria caso os pedidos de resgate fossem realizados em tempo real, quando o cliente se apresentasse no local do evento. Outras soluções mais sofisticadas podem ser exploradas, nomeadamente capitalizando em melhorias de desempenho que venham a ser conseguidas na tecnologia de *blockchain.*

A minha envolvência no projeto desde o início, e em todas as suas fases, permitiu-me desenvolver uma análise crítica sobre quais os casos de uso que poderiam vir a beneficiar com a utilização de uma *blockchain*.

Palavras-chave: Blockchain, Smart E-Tickets, Criptomoeda, Hyperledger Fabric, Confiança

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

This chapter presents the problem addressed in this work, developed in Accenture. It also discusses the approach chosen to try to solve it and the motivation, objectives and the context of development of the work.

1.1 Problem

With the number of spectators attending events increasing, this growth is also followed with the increase of people trying to resell tickets to these events for prices much higher than its original price. As reported by *Instituto Nacional de Estatística* [1], tickets revenue increased by 42,6% to a total amount of 85 million euros in 2016 and the number of spectators increased by 18,8% to 14,8 million. When someone resells a ticket for a price higher than its actual price it is going against the Portuguese law, that says that if someone sells any kind of asset for a price above its original price can be punished from 6 months to 3 years in jail and a be fined with at least 100 days [2]. As reported by *Expresso* [3], the *Autoridade de Segurança Alimentar e Económica* caught 24 people trying to resell tickets for prices between 150 and 1000 euros when their original prices were between 37 and 338 euros, this represents a profit margin above 900% in some of these cases. This is a problem that affects everyone involved with events, from the authorized resellers to the people who just want to buy tickets and to the government, because when people are selling tickets in the conditions described above, they are making money from ways that are not legal and avoiding paying taxes for each ticket that they sell. When people buy tickets from these resellers, they cannot be sure that that ticket is authentic and when they get to event entrance, they cannot be sure they will be able to attend the event that they bought the ticket for. There are two ways of acquiring a ticket today, or the ticket is bought in the physical store or the ticket is purchased online and sent to the buyer's email. Sending the ticket to the buyer's email address obliges the buyer to print this ticket and bring it to the event. There is more than one thing that can go wrong, the ticket can be lost, stolen, not valid, and there is nothing that can be done when one of these things happen, the buyer is always the one who's going to lose.

Technology can assist with this problem, starting the transition from the traditional paper ticket that is necessary to take to event so that it can be scanned, to a digital ticket that does not need to be printed, can be verified if it is authentic, has an owner assigned to it and the only way to resell this ticket is to make a transaction within a platform that will be created to assist with these trades.

This work focus on this problem and aims to develop a solution that can assist authorized resellers and event organizers to sell their tickets and have more control over these tickets.

1.2 Approach and Motivation

The approach chosen to be developed for a possible solution for the problem described, consists of developing a web platform that works as a marketplace for event organizers and authorized resellers to be able to sell the tickets for the upcoming events, and for everyone else to be able to buy tickets for these events. As the online ticketing platforms facilitate the way people buy tickets, it made sense for the approach to be based on a web platform. The focus of this work is to explore current blockchain technologies and how they can help with the problem.

As an emergent technology, blockchain has been described as a technology that can revolutionize several industries, including Financial Services, Supply Chain, Healthcare, Real State, and Politics. For Accenture, who wants to be in the front line of innovation, being able to have collaborators who can acquire skills and knowledge in the new technologies, like blockchain, and then instruct other with the same skills, is one of the goals that a company like Accenture must have.

As a software engineering master's student, being able to have an opportunity to work with a technology like blockchain, the technology that allowed Bitcoin and other cryptocurrencies to grow and gain popularity the last couple of years, research and develop a use case that I am particularly fond off, is one of the main motivations to enrol in this project. Blockchain can remove the need for third parties to be involved in transactions, allowing people to save money because they will not need to pay third parties a fee for the work that they do. For this project, learning about blockchain and develop an application with one of the blockchain platforms that are currently available as open source, allows for an understanding of how these applications are developed, and as a future software engineer to develop well-structured applications and how these can be improved.

The ticketing industry nowadays still mainly works with the traditional paper tickets which are often sent by email to be printed by the buyer. For example, if the ticket is bought through the authorized reseller online platform, in some cases there is no need to register in the platform and in this case the buyer's information is given just to add his name to the ticket and be possible to send the tickets to his email. In these cases, there are several things that can go wrong, in particular the tickets can be lost. If something happens to the email account and the tickets were not downloaded or printed before, the tickets from that point on are inaccessible. The most important is that, there is no option to resend the tickets and the consumer is the one that loses. If there is an option to resend the tickets, these can be used to increase ticket fraud by reselling the same authentic ticket several times. It is necessary to have a method to map each ticket to his owner and remove at the same time ticket fraud and secondary market ticket business. When the ticket is bought in the physical store, this ticket will be printed and handed to the buyer. If this ticket is lost, there is no backup ticket available to replace it, and the consumer will lose the ticket and the chance to attend the event.

The traditional ticketing industry methods are shown faulty and in need to be improved. It is crucial for consumers to have trust when buying tickets, and nowadays they have no other alternative but to trust in the authorized resellers. When buying from well-known authorized resellers, this problem does not appear to be an issue. However, with the new sellers emerging, namely international sellers, an issue like this could mean the loss of clients for these sellers. Trust comes to be the most important factor when deciding where to buy tickets, and event organizers need people to attend the events and authorized resellers need for people to trust them.

The world of cryptocurrency emerged with bitcoin, and since its creation the number of cryptocurrencies grew to more than 1600. Since blockchain is the technology that supports cryptocurrency, by being digital, cryptocurrencies cannot be counterfeited or reversed arbitrarily by the sender, contrarily to the regular credit card payments, which can be reversed using chargebacks.

The ease of transferability of the cryptocurrency between people or through an established cryptocurrency exchange, is one of the advantages of this technology. By being a technology that assists decreasing fraud, it fits with the ticketing industry current situation.

1.3 Objectives

The objectives for this work are to develop a web platform that allows for users to buy tickets from authorized resellers, resell legally these tickets according to the law and to allow authorized resellers and event organizers to sell tickets through the platform. At the same time, the aim is also to get knowledge about the technology of blockchain, the frameworks and platforms that are currently used to develop decentralized applications.

1.4 Context

This work was developed from 12 of October of 2018 to 19 of July of 2019 in Accenture. Accenture is a global professional services company, that provides a broad range of services and solutions in 5 different areas, they are Strategy, Consulting, Digital, Technology and Operations. Accenture can provide experience and specialized skills across more than 40 industries, with more than 450,000 people serving clients in over 120 countries [4]. Since they are a company that wants to be ahead of the others, this project fits with Accenture Innovation [5] overview of being able to help their clients to develop and deliver disruptive innovations.

The elements working with similar goals as myself, were Miguel Andrade and João Leal. We were all doing an engineering project with Accenture as part of our master in Informatics Engineering, but each of us were developing a different use case with the goal of explore current blockchain technologies and how they can help with each use case identified by us. We had as supervisor and the responsible for the project Tiago Minchin and as day-to-day supervisor Álvaro Silva.

With the goal of exploring current blockchain technologies, at the beginning of this work it was necessary to research what were the advantages and disadvantages of this technology and research for use cases in distinct industries where blockchain could be beneficial.

1.5 Structure of the Document

This report will be structured as follows: Chapter 1 has the introduction to the problem that will be the focus of this project and where it is presented the objectives and motivations, from the point of view of Accenture and the point of view of the student. Chapter 2 it is composed by the state of the art and an introduction to what blockchain is, the current platforms for development and the methodology of software development that is going to be used. Chapter 3 has an overview of the solution and the process that lead to the decision to work the use case present in this work. Chapter 4 has the design and implementation of the system, along with how the blockchain network was setup and deployed. Chapter 5 concludes the report with an analysis of the work developed, limitations encountered of what it was proposed at the beginning of this work and what could be developed in the future to improve the use case worked on.

Chapter 2. Context and Related Work

In this chapter will be discussed the state of the art of blockchain, with an introduction to blockchain and its key characteristics, some of the available platforms to develop applications with blockchain technology, cryptocurrency, the methodology for software development chosen and related applications that currently exist.

2.1 Blockchain

Blockchain is a new type of distributed database system that maintains and records data in a way that allows multiple stakeholders to confidently and securely share access to the same data and information. This database is shared across a network of computers and after a record has been added to the chain it is very difficult to change any of the data that was recorded, once this data is recorded it stays in the chain forever, after being validated. Each transaction is recorded and stored in the blockchain and these records can have any kind of information, that are bundled together into blocks and added to the chain in which each block knows about the previous block. Each block upon its creation has a unique cryptographic hash that is created by a mathematical function that receives digital information and generates a string from it. When a block is generated it contains a hash that is generated by a cryptographic function and each block contains a *nonce*, the hash of the previous block, a timestamp, a *Merkle tree hash* [6] and a list of all the transactions in this block. This means that if someone tries to change any information from a block, breaks the chain because the next block will still have the old hash, and so on. Recalculating the new hashes for the whole chain can be an impossible job. The amount of computing power needed to recalculate all the hashes in the next blocks is too big to be performed by a simple computer in a reasonable amount of time.

The simplified block structure is represented in the Figure 2.1. Every blockchain network needs a block from it starts, this block it is called *Genesis Block*, it defines the initial parameters of the blockchain network.

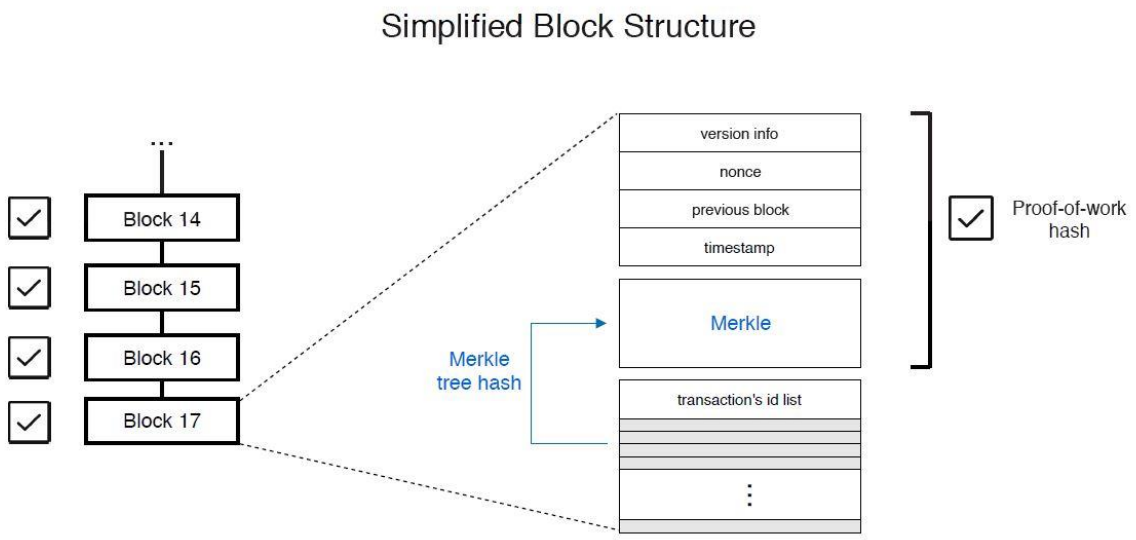

Figure 2.1 - Simplified Block Structure

Before a block can be added, there is a *consensus mechanism*, that verifies if the information being added is valid, the network must be in consensus. This mechanism allows for the next block being added that represents the most current transactions in the network. Among the existing consensus mechanisms, the most popular are the Proof of Work (PoW) and Proof of Stake (PoS). In the PoW mechanism, participants are required to solve an expensive computer calculation to be able to create a new block — this is called mining. Upon solving the calculation, the *miner*, who solved the calculations, is rewarded. For example, on the Bitcoin network, a miner is rewarded with bitcoins. In this mechanism, there is a network of miners that compete to be the first to find solutions for the expensive calculations. In the PoS mechanism, it is chosen deterministically who is going to be the creator of the next block, depending on its wealth. If some participant has 1% of the wealth on a blockchain, it can mine 1% of the PoS blocks. Contrarily to PoW, there is no block reward when a block is mined, and instead of calling miners to the creator of a block, they're called *forgers*. PoS systems are much more cost-efficient and greener than PoW because the computational power required to operate each one is much smaller. A PoW system consumes much more energy than a PoS system.

The Figure 2.2 exemplifies how blockchain works.

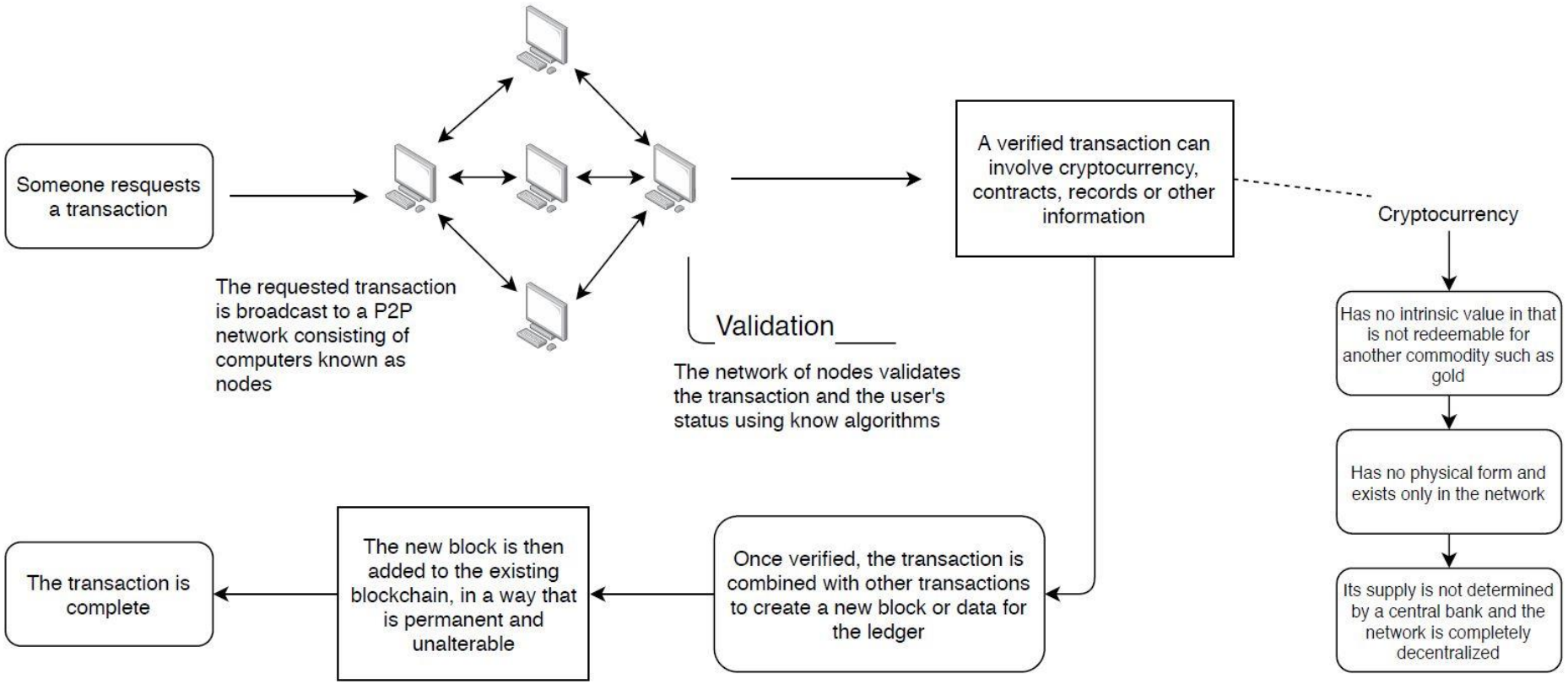

Figure 2.2 - Blockchain in action

2.1.1 Types of Blockchain

There are different types of blockchains: public, private and consortium. Each come with different advantages and the type of blockchain chosen should fit the project needs. According to the National Institute of Standards and Technology [7], blockchain can be categorized based on their permission model, *permissionless* or *permissioned*. In a permissionless blockchain network, anyone can publish a block and in a permissioned blockchain network only certain users can publish blocks.

Public

A public blockchain, like Bitcoin and Ethereum, can also be referred as a permission-less blockchain. Within this type of blockchain, everyone can be a part of the network and can be an active participant by running as a node. This allows for the mining of blocks and by making transactions in the blockchain.

Private

A private blockchain, like Hyperledger, can also be referred as a permissioned blockchain. For this type of blockchain, only authorized participants can be a part of the network.

Consortium

A consortium blockchain, like Quorum and R3 Corda, are defined as being partially decentralised. It is managed by a group of organizations instead of being managed by one organization as in private blockchain. The member organisations have the authorities to participate by running as full node, by mining and making transactions within the blockchain.

2.1.2 Smart Contracts

A Smart Contract is computer program that directly and automatically controls the transfer of digital assets between the parties under certain conditions. A smart contract is a digital improvement to the regular contracts, in a way that the smart contract forces the contract to be executed under the conditions in it defined. Smart contracts are pieces of code, programmed accordingly to the contract conditions that were defined in the business logic. The idea behind smart contracts, are that the it works based on the if-then logic, if someone sends me an object, then the sum of money will be transferred to who sent the object. They are embedded in the blockchain, which allows for transparency, immutability and decentralization of the contract itself. The way applications communicate with these contracts is by using its address.

Some of the advantages of smart contracts include record keeping, by storing all transactions in chronological order in the blockchain to facilitate audits. Eliminates the third parties, allowing for a more transparent and direct communications between clients and organizations.

Smart contracts also have its disadvantages or limitations, like it is still prone to human error because it is still necessary programmers to code these contracts. Whenever a change is made, it is necessary to create a new contract and implement it in the blockchain. Being such a new concept, there are no standards to help with its implementation.

Platforms like Bitcoin, Ethereum, R3 Corda and Hyperledger allow for smart contracts to be used in their platforms. Within the Hyperledger platform, smart contracts are called Chaincode that runs in a secured Docker container. Smart contracts in platforms like Hyperledger handle business

logic agreed by every member of the network. In platforms like Ethereum, because it is a public blockchain, anyone can deploy its own smart contract to the blockchain.

2.1.3 Blockchain Platforms

With an increasing interest for blockchain and projects that want to take advantages of this technology, the need to have blockchain platforms that allow developers to develop blockchain applications with ease is higher. These blockchain platforms have their own properties and advantages and need to be chosen according to the project's needs.

Ethereum

Ethereum is an open software platform created by Vitalik Buterin, Gavin Wood and Joseph Lubin, that takes blockchain technology and allows developers to build and deploy decentralized applications. Ethereum has its own cryptocurrency called *ether*, which is one of the most known cryptocurrencies and that can be traded for other cryptocurrencies and sovereign currencies. Ethereum is a type of public blockchain network but it also allows one to create private networks. It comes with the possibility to create smart contracts, written in Solidity, and a step-by-step process to create new cryptocurrencies. [8]

Hyperledger

Hyperledger comes in many frameworks and tools to allow developers to develop their own applications. Hyperledger is an umbrella project of open source blockchains and related tools. It started in December 2015 by the Linux Foundation. Between their frameworks, Hyperledger Fabric is one of the their most used framework platforms and it is a type of private blockchain platform to be used in enterprise contexts. Fabric also has smart contracts, which are called *chaincode*, and these contracts can be written in Java, Go and Node.js. [9][10][11]

R3 Corda

Corda is an open source blockchain platform created by a consortium of the top banks in the world. Corda, like Hyperledger, is a type of consortium blockchain with pluggable consensus. It supports multiple consensus providers employing different algorithms on the same network. For smart contracts it takes advantage of Java and Kotlin programming technologies. [12]

Quorum

Quorum is an enterprise-focused version of Ethereum developed by J.P. Morgan and is one of the first adoption of blockchain technology among financial industries. Quorum differs from Ethereum on privacy. It supports private transactions and private contracts through public/private state separation. With no need for PoW or PoS consensus in a permissioned network, Quorum in alternative offers multiple consensus mechanisms that are more appropriated for consortium chains, like Raft-based Consensus [13] and Istanbul BFT [14] [15].

2.2 Cryptocurrency

The first cryptocurrency was first introduced in 2008 in a white paper wrote and published under the pseudonym "Satoshi Nakamoto", this cryptocurrency was Bitcoin [16]. Bitcoin was developed with the intention to allow people stop relying on third-party mediators, like banking institutions. After the global financial crisis of 2008 and 2009, people's trust in the banking institutions and government's abilities to have control in the financial industry decreased. As defined by Luther [17], "*Cryptocurrencies are digital alternatives to traditional government-issued paper monies*", that are not issued, controlled or backed by any government. Cryptocurrencies come with the advantages of being a technology that is decentralized, there is no central authority in the network, the network is distributed to all participants and the network never really goes offline. It is anonymous and transparent as the information that is transacted, and the owner of this information is store on separate blockchains. The speed of transactions that allows for someone to send money to anywhere in the world in a matter of minutes, after the transaction has been processed by the network. These transactions do not require for any personal data to be disclosed, it uses a private and public key to encrypt and sign the transactions. This does not leave space for fraud, only the public key is available for everyone and the private key only the owner has access to it. Each transaction needs to be signed by the owner with its private key and then it is applied a mathematical function, hash. This grants that the transaction is only performed by the owner, he is the only one that has access to his private key. Cryptocurrencies also have disadvantages, one that is seen every day is its value. The up and downs of their value, creates large risks for someone to invest in it in the medium and long term. This volatility is caused by the statements that are done by the governments of different countries. Being a young technology that is still under development, it is still prone to the difficulties and risks that outcome of its initial growth. [18] [19]

2.2.1 Cryptocurrency vs Token

Although cryptocurrency is the most common name given to every type of digital currency, cryptocurrencies are categorized in two ways: alternative cryptocurrency coins or just coins, and tokens. These coins are usually alternatives to already existing coins, like bitcoin, where these coins were created by modifying bitcoin's source code, like Dogecoin. Not all coins were forked from bitcoin's source code, Ethereum and Ripple have their own native cryptocurrencies. Tokens are a representation of an asset or utility, that usually resides on top of another blockchain. This token can represent any assets that are tradeable, from commodities to loyalty point or event other cryptocurrencies. When comparing the complexity to create a token or a cryptocurrency, creating a cryptocurrency is more expensive than a token, it is easier to create a token and most of the times what is needed it is a token and not a cryptocurrency. In February 2018, Swiss Financial regulators FINMA published guidelines that defined asset, utility or payment tokens [20]. They defined payment tokens as "synonymous with cryptocurrencies and have no further functions or links to other development projects. Tokens in some cases only develop the necessary functionality and become accepted as a means of payment over a period of time". Utility tokens as "tokens which are intended to provide digital access to an application or service". And asset tokens "represent assets such as participations in real physical underlyings, companies, or earning streams, or an entitlement to dividends or interest payments. In terms of their economic function, the tokens are analogous to equities, bonds or derivatives". In a public blockchain, the participants are general public and it is necessary to provide

incentives, like cryptocurrency, for people who help maintain the state of the blockchain. In the private blockchains, the participants are not general public and they are involved in a business, there is no need for incentives to be used.

2.3 Ticketing Industry Landscape

The ticketing industry problem has been and is still trying to be addressed through the development of applications that will decrease exorbitant secondary market prices, control who gets to resell tickets, stop scalpers from making a business out of the tickets that they purchase, improve tickets authenticity and trust from people in the tickets purchased and decrease tickets fraud. Herein, we briefly discuss five of the solutions that are addressing these problems.

GUTS Tickets

GUTS [21] is a blockchain ticketing system that registers ownership of tickets and uses its own protocol called GET, that offers features for ticking and booking for events and promising to end with exorbitant secondary market ticket prices and ticket fraud occurrences. GET is also the name of their cryptocurrency that is used to buy and sell tickets within their app. GUTS is built on the Ethereum platform.

GUTS is a solution that uses blockchain technology that has been more successful than others by selling around 120000 tickets for a single event. Using a public blockchain platform, Ethereum in this case, comes with the cost that is necessary to pay for transactions and the limitation of transactions per second that the platform has. At this date, Ethereum is handling 15 transactions per second, which is an obstacle for applications based on the ticketing industry that needs to validate a high number of tickets per second per event. In a public blockchain network, applications use cryptocurrencies to be able to finance itself and pay for the infrastructure, by having an initial coin offering which is how funds are raised for a new cryptocurrency offering. GUTS validate tickets off-chain by using a database, Postgress, to improve the validation speed of each ticket. This is necessary due to current blockchain limitations, and the requirements that come with the characteristics of an event. One of GUTS principles is to use a dynamic Code, like a QR code that changes as a function of time and when the ticket is transferred between users, that is only revealed before an event starts. This is one of the ways that can prevent scalpers from selling fake tickets because this code will not be available when they are trying to sell the tickets.

Aventus Protocol

Aventus Protocol [22] is a blockchain based event ticketing solution to help eliminate fraud and unregulated touting. This is done by allowing for the event organizers to exert more control over their inventory and processes. Rights holders can define rules across the ticketing supply-chain. Including promoters, venues primary and secondary agents, to which everyone must adhere. Aventus Protocol is built on the on the Ethereum network.

Aventus at its core is a model where off-blockchain transactions of tickets are prevent, by having all tickets sales going through a platform and defining tickets resale price range. While GUTS have its own platform, events are listed on the Aventus protocol and will be accessible to any application that uses the protocol. Their solution associates each ticket with a photo that identifies the owner of the ticket. This photo can be just the owner’s face, an identity document or a credit card that

is added when the ticket is purchased. This identification can only be changed when the ticket is resold through a secondary market that uses the Aventus protocol. When the tickets are resold, the tickets owners don't know who they are selling the ticket to.

Upgraded Tickets

Upgraded [23] is a solution that converts traditional event tickets into secure interactive digital assets that are protected by blockchain. Upgraded uses encrypted barcodes to prevent tickets fraud currently existing in the traditional paper tickets. This solution was acquired by Ticketmaster, trying to improve fans and artists experience when buying and selling tickets by preventing fans from buying fraudulent tickets and giving more control over tickets distribution to the event organizers. Upgraded works on the Ethereum platform.

Ticketline

Ticketline [24] is a web platform, acting as intermediary, that sells tickets for events. This platform is one of biggest tickets selling platform in Portugal and where it is possible to find tickets for almost every event happening in the country. Their processes of distributing tickets after they were purchased on their platform include, sending the tickets through the post offices, Ticketline headquarters or on the day of the event at the venue. The electronic ticket is sent to the buyers' email as a pdf file, with a bar code associated.

TicketSwap

TicketSwap [25] is a web platform to buy and sell secondhanded tickets for events like concerts, festivals, sports events and theatre where the sellers on the platform are verified based on several criteria. Sellers upload their tickets and provide personal information, name, bank account number, phone number and email, required by the platform. This personal information is required because in the case of the seller committing fraud, TicketSwap will be able to contact the seller, ban him from the platform and provide all the information needed to file a police report. Every ticket is also checked in multiple ways, location, price, correct date, barcodes and others, to ensure the uniqueness of the ticket being sold on the platform. When a ticket is sold, it is made invalid and the buyer receives a new and unique ticket, this is what they called Secure Swap. By partnering with ticketing companies, they can validate the ticket being sold. The ticketing company invalidates the seller's ticket and sends TicketSwap a new ticket, which is then sent to the buyer. Listing a ticket on the TicketSwap platform is free, but they charge a 5% service fee at both buyer and seller. It is allowed for tickets to be sold 20% above its face value. TicketSwap is available in several countries, most of them are in Europe but they are also available in the US, Canada, Australia and New Zealand. They have been partnering with Sziget Festival since 2017, one of the largest music and cultural festivals in Europe, more specifically in Hungary. With an estimated 565.000 of visitors in 2018, these numbers attract scalpers and other type of fraudsters, attempting to deceive the fans to want to attend the music festival. They have been also partnering with Paylogic, a ticketing technology company and Ancienne Belgique, Belgium's premier mid-sized music venue.

	Blockchain	Blockchain Infrastructure	Token/ Cryptocurrency	Governance	Resell Tickets
GUTS Tickets	Yes	Public, Ethereum	GET	Closed source	Yes
Aventus Protocol	Yes	Public, Ethereum	AVT	Open source	Yes
Upgraded Tickets	Yes	Public, Ethereum	-	Closed source	-
Ticketline	No	-	No	Closed source	No
TicketSwap	No	-	No	Closed source	Yes

Table 2.1 - Existing Solutions Comparison

2.3.1 Smart E-Tickets

The solutions described previously, range from the current online selling process of tickets to the ones who are incorporating blockchain technology to improve the trust in the tickets bought. The solution proposed in this work pretends to be a marketplace where different event organizers can create and sell tickets for their events. While solutions like Ticketline do not allow for the reselling of tickets by the client, it just focus on the process of selling tickets. Not giving an alternative for the client to resell his ticket if he cannot attend the event, leads to the reselling of ticket in the secondary market which there is no control over the price of the tickets. The marketplace in this work, wants to facilitate the process of reselling tickets, by giving the client the option to resell his ticket in the marketplace. This option grants a greater control over the reselling of tickets to the event organizers and avoids that clients sell and buy tickets in the secondary market for outrageous prices.

Solutions like GUTS Tickets and Aventus Protocol, have their solutions based on public blockchain platforms. For this companies to support the costs of the infrastructure that allow them to continue operating, they must have a cryptocurrency. The common type of funding using cryptocurrencies is called Initial Coin Offering, ICO. By using a private blockchain, like IBM's Hyperledger Fabric, the stakeholders support the cost of the infrastructure, removing the need for a an ICO. The use of a token in the applications that use a private blockchain platform are not required, it only may assist with payment and transaction of assets within the application if desired. The solution proposed uses IBM's Hyperledger Fabric, that needs the stakeholders involved to support the infrastructure, so that the online marketplace network may operate.

Chapter 3. Solution Overview

This chapter starts by presenting the process carried out for the selection of the use case for this work, since the identification of several use cases to the use case that was selected. Then, it provides an overview of the proposed solution and its context.

3.1 Use Case Selection Process

As gaining knowledge in blockchain was the main focus of the company, we had to research possible use cases that would be impactful in its industry. Based on a previously elaborated use case evaluation matrix by the company, we identified several use cases, some with a higher feasibility than others. But nonetheless, it helped us at the beginning to identify a broader range of use cases. Therefore, we were involved in the process of identifying, evaluating and defining the use case that would be addressed this work would consider.

3.1.1 Identification of Use Cases

With the process of defining a use case that would benefit with the use of blockchain technology, we started by researching on internal company documents what has been done around blockchain. With the information that was gathered we identified some possible use cases. These were complemented by researching, on the internet, current applications of blockchain being explored by the blockchain community.

The use cases that were identified cover a couple of industries where blockchain could be impactful. The industries identified and some of their possible applications were as follows:

- Industry of Communications, Media and Technology: the register of ownership rights of music, film and television;
- Health Industry: storing of medical data from patients on the blockchain for a higher control of who can access the patient's information by having the patient have full control of its data;
- Products Industry: use cases for warranties, by going from a paper warranty for a digital warranty. Cars and airplane maintenance tracking record, luggage tracking, decentralized game vault, which would allow storing videogames keys in a secure way and improve the ticketing industry with e-tickets that use blockchain, allowing for a more transparent industry and decrease the exorbitant secondary market prices;
- Public Services Industry: would use blockchain to store citizens driver's license, their drive's history and relevant information associated with their driver's license;
- Resources Industry, record farmers quota and ensure they are paid according to cooperatives' rules;

- Sports Industry: give more control over ticket sales, create a currency to allow fans to buy tickets, merchandise, and receive currency due to fans loyalty to the club;
- Financial Services Industry: create a blockchain solution to improve the current, outdated and time-consuming, cheque clearing and credit settlement system.

Within these industries some of the identified use cases were considered difficult to implement while others would require less work but could only be implemented as a proof of concept.

3.1.2 Evaluation of Use Cases

With several use cases identified, the process of evaluating them with use of a matrix, allowed us to reduce the number of use cases that could be chosen for this work. The drivers that were used to evaluate the use cases were:

- Project interest;
- Ease of implementation;
- Relevance to company's strategy or industry;
- Blockchain relevance for the use case;
- Legal/Regulatory/Compliance aspects and complexity;
- Number and type of actors involved;
- Interoperability with existing systems and third-party integration;
- How open/favourable the internal or external business partners are;
- Market exposure.

These drivers allowed us to filter the initial use cases into a small set of possible use cases to be worked on. The evaluation of the use cases was made with quantitative values from 1 to 4, 1 being the lowest value and 4 the highest value of agreement.

The small set of uses cases that were defined as being the most attractive to work on were:

- Etickets;
- Healthcare;
- Cheques;
- Pensions;
- Insurance.

The evaluation of the use cases was made based on information gathered on each type of industry that each use cases fits in and know-how available internally. At the end, one of the drivers that influenced our decision to choose the use cases was the interest in the project.

A comparison of a small set of use cases that had the most potential is represented in the Figure 3.1 - Use Cases Comparison. From this small set of use cases, the etickets, healthcare and cheques, were the use cases that were chosen to develop proof-of-concepts that could improve the current process of each use case. This work addresses the etickets use case, focused on the selling and reselling of tickets using blockchain technology.

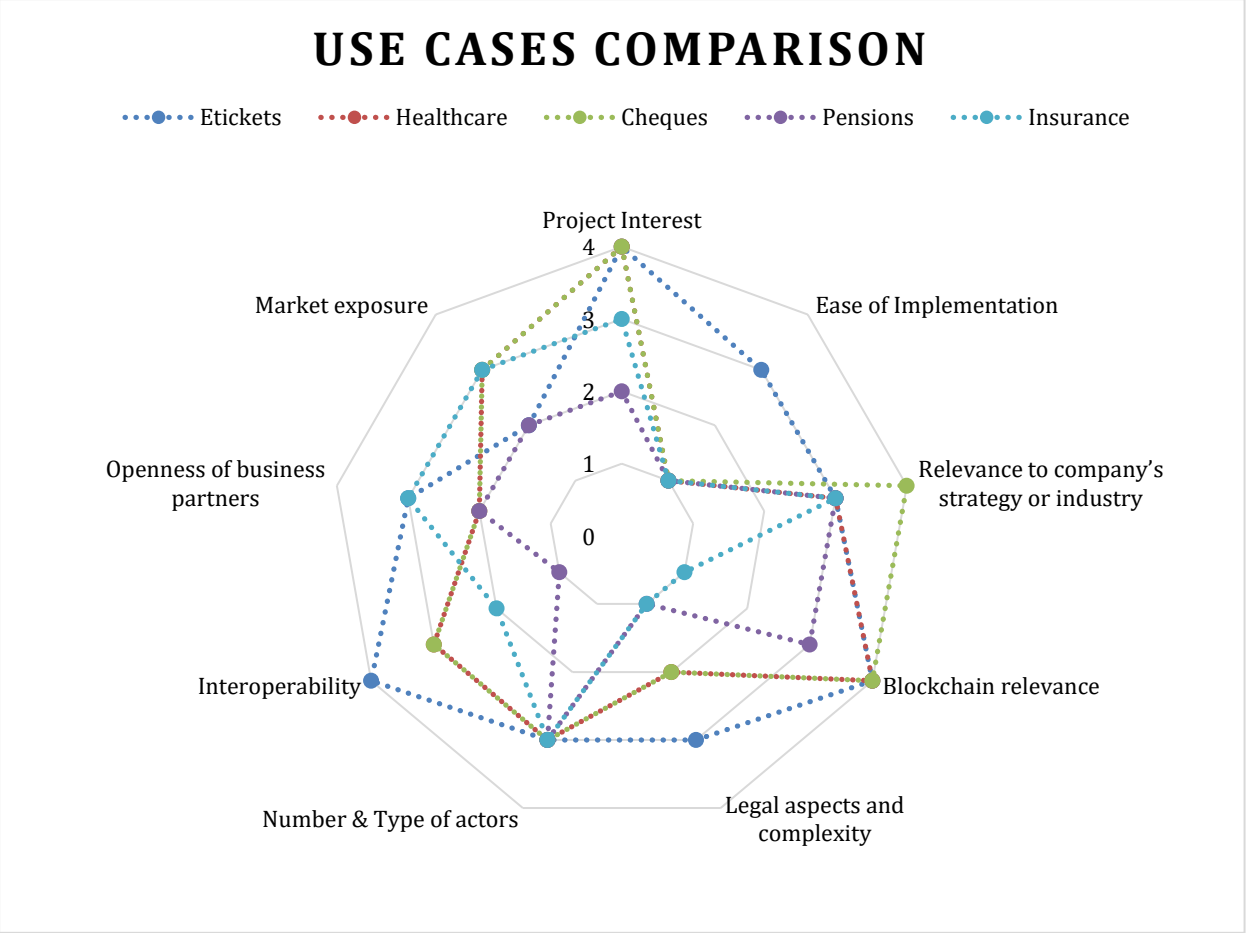

Figure 3.1 - Use Cases Comparison

3.2 Solution

The solution proposed in this work is to create a web platform that would act as an online marketplace, where event organizers can sell the tickets for their events, have more control on who gets access to resell their tickets and prevents the reselling of tickets for exorbitant prices. The solution does not intend to remove the middleman completely. Instead, the solution supports that authorized resellers resell tickets for events and create event packages. Even though one of the advantages of blockchain technology is that the middleman is not necessary anymore, keeping the middleman in this use case adds value and increases the promotion and advertising of the events through the authorized resellers.

An authorized reseller adds value to the business because they have an established customer base that trusts them. They are better prepared to sell the tickets because they have the know-how on how to sell the tickets effectively. They are already prepared on how to market the events in the most efficient way and already know their customers because they know how to handle the selling of tickets and know which services they need to provide to their customers.

An online marketplace allows for businesses to sell their products with a relatively low cost and provide great opportunities for their business. With events often captivating international attention, an online marketplace allows for people from other countries to purchase tickets for events happening in

a different country. Having multiple authorized resellers selling their tickets in the same place, offers a convenient way to compare ticket prices and offers greater transparency regarding the price of the tickets between the different authorized resellers.

As authorized resellers are an important part of the current process of the resale of tickets, we want to keep them and have them still being involved in this process. When an event organizer creates an event and sets the dates, the tickets sales will not be available for the general public at first. The initial sale period it is for the authorized resellers that have an agreement with the event organizers. This means that the authorized resellers can buy a batch of tickets for them to sell. The face value for each ticket is unchangeable, which means that the authorized resellers would buy each ticket for a lower price than its face value. The price for each ticket in the batch would be negotiated between the event organizers and the authorized resellers.

The tickets are available to the clients when the event organizers and/or the authorized resellers set them for sale. After this, the clients can purchase a limited number of tickets with their account and these tickets will stay associated with the client who bought the tickets. The client can have two roles in the marketplace: as the person who wants to purchase tickets and as the person who wants to resell previously purchased tickets. The face value of the ticket never changes, so when the client puts its ticket for sale, the price of this ticket that is being resold is at the same price that it was bought. The tickets for the events are only available and can only be traded in the platform. It is not possible to trade the tickets outside the platform.

At the day of the event, ticket transactions would close hours before the event starts. With the transactions terminated, a sold ticket cannot suffer more changes and its owner would be definitive. This allows for the unload of all the tickets, along with their unique identifiers, to an auxiliary database, that it would be used by the venue responsible for the validation of the tickets. The unload of the tickets to an auxiliary database, enhances the verification speed than if it were used blockchain technology. A client would just need to access the platform, select the respective ticket for that event and allow the venue staff who is scanning the tickets, to scan his ticket. For this verification, the venue staff would use an event scanning application or machine, that would interact with the auxiliary database and validate the tickets.

The main problem that this solution addresses is the resale of tickets. Scalpers and black-market sellers do not bring any value to the Ticketing Industry, instead they bring mistrust and fraud to the whole process of reselling tickets. Reselling tickets at prices above the ticket face value is punishable by the Portuguese Law even so this continues happening regularly. One of the problems is that most tickets still need to be in paper format for it to be used. This leads to fraud and reselling the same ticket multiple times, with the general public being affected. The goal is to overcome these problems and improve the process of reselling tickets.

3.2.1 Context

An online marketplace has impact on different types of stakeholders, directly and indirectly. It gives an easy way of comparing prices between authorized resellers and event organizers and allows for the tickets to be in a digital format and not being necessary for them to be printed. With different stakeholders involved, the use of a blockchain makes it possible to maintain a history of changes for each ticket that allows for a greater control over the tickets for each event.

The problem described before, regarding the reselling of tickets, is addressed in this work by migrating the traditional paper ticket for a full digital solution. By avoiding the necessity to print

tickets to attend an event and having a technology that does not allow for the creation of duplicated tickets and supports the tracking of the changes that a ticket might suffer during its lifecycle, we will improve the current process. With a digital asset and with blockchain, the possibilities to have duplicated tickets are low due to the database replication and computational trust, any alteration to one of the machines' database is caught during the transactions. Since blockchain promotes the decentralization of authority, the stakeholders must trust each other's and the smart contracts running in the network ensure the enforcement of the business logic.

Event organizers, authorized resellers, clients and venue staff are the stakeholders involved in the process of selling, reselling and redeeming tickets. The marketplace uses the payment system as a mean to convert the money to tokens and tokens to money. The tokens are the currency in the platform that clients use to buy tickets from event organizers, authorized resellers or other clients.

The following Figure 3.2 represents the context diagram of the system.

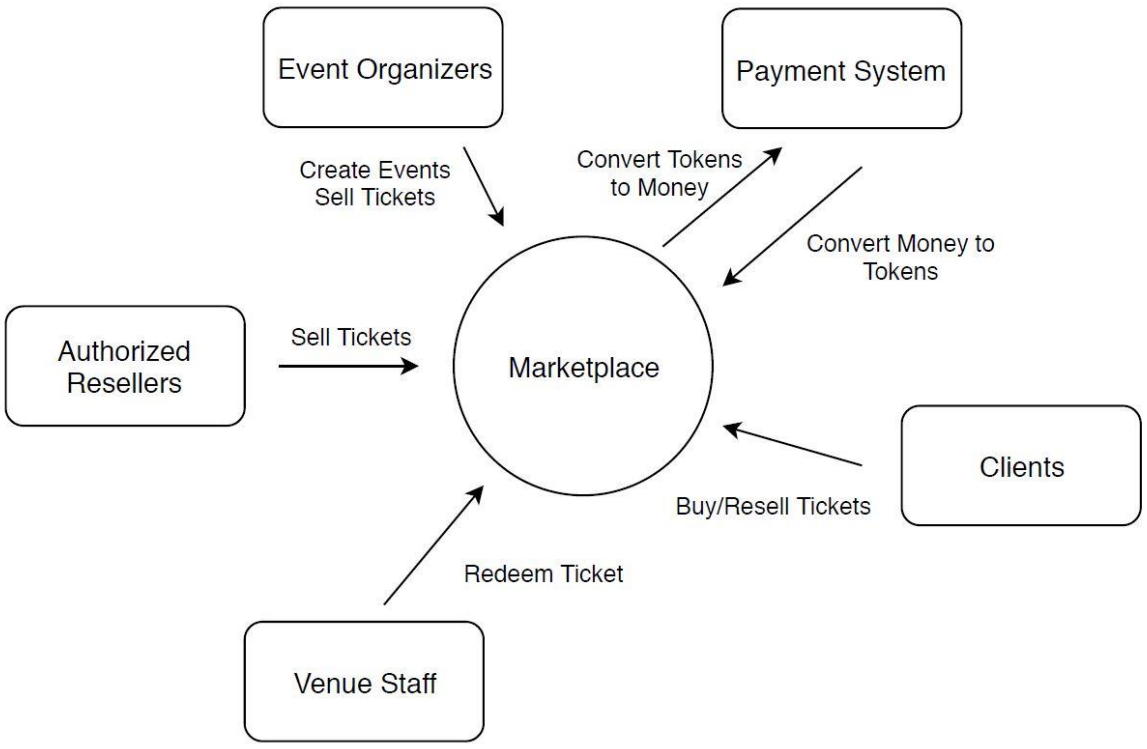

Figure 3.2 - Context Diagram

Event Organizer: On the proposed solution, the event organizer accesses the web platform and registers the event along with its essential information. Once an event is created, the organizer can create the tickets for the event and handle himself the ticket sales or he can reach an agreement with multiple authorized resellers and have them sell the tickets. If the organizer wants to manage and have full control of the sales process, he just needs to create the tickets and put them for sale. When a ticket is created, the event organizer is set has the owner of the ticket. If he wants the authorized reseller to sell his tickets, they can sell a batch of tickets to the authorized reseller for a lower price per ticket but with the face value of the ticket unchanged. With this, the organizer can sell a high number of tickets at the time to multiple authorized resellers and have them manage the ticket sales.

The process flow of creating an event by the event organizer in the proposed solution is:

1. Event Organizer wants to organize an event
2. Organizer access the platform
3. Create and fill event information
4. Organizer wants authorized reseller to sell his tickets?
    a. Yes, contact authorized reseller and reach an agreement for the price of the batch of tickets
5. Create tickets and the Event Organizer is set has the owner of the tickets
6. Tickets available for clients to buy

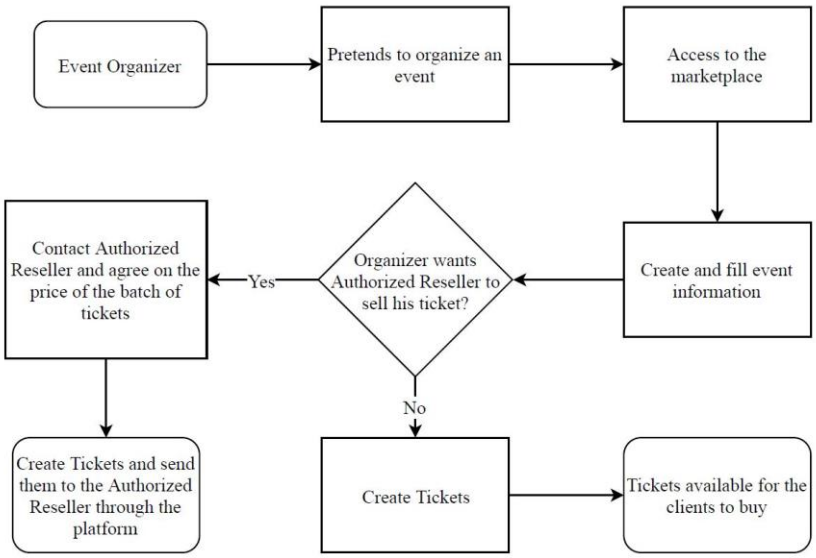

Figure 3.3 - Event Organizer Process

Authorized Resellers: The way it is done online today, the authorized resellers have their web page where they sell tickets for the events that they are authorized for. For a person who wants to buy a ticket for an event, he must access the web page and search for the event. In some cases, it is necessary to have an account to purchase tickets and, in other cases, it is just necessary to provide essential information when the process of buying the tickets is in progress. In the proposed solution, it is necessary that event organizers and authorized resellers are registered in the platform, for being able to register future events, to register tickets and to sell tickets for these events.

Authorized resellers can have privileges for the purchasing of tickets when they have an agreement with the event organizers. If the event organizers still want for the authorized resellers to be a part of the process and promotion of the event, they can reach an agreement for what they are going to pay for each ticket. Despite this, the face value of the ticket does not change, the authorized resellers can purchase a batch of tickets for a lower price per ticket, but they still have to sell them at face value.

The process flow of selling tickets by the authorized reseller in the proposed solution is:

1. Event Organizer wants an Authorized Reseller to sell his ticket

2. Authorized Reseller gets contacted by the Organizer and negotiate the price of the batch of tickets
3. Agreement reached?
    a. Yes, authorized reseller pays for the batch of tickets with tokens on the platform
    b. No, renegotiate or contact another authorized reseller
4. Authorized Reseller receives batch of tickets for the event from the Organizer and the tickets ownership is transferred to the Authorized Reseller
5. Promotes, creates special offers, sells and manages tickets for the event

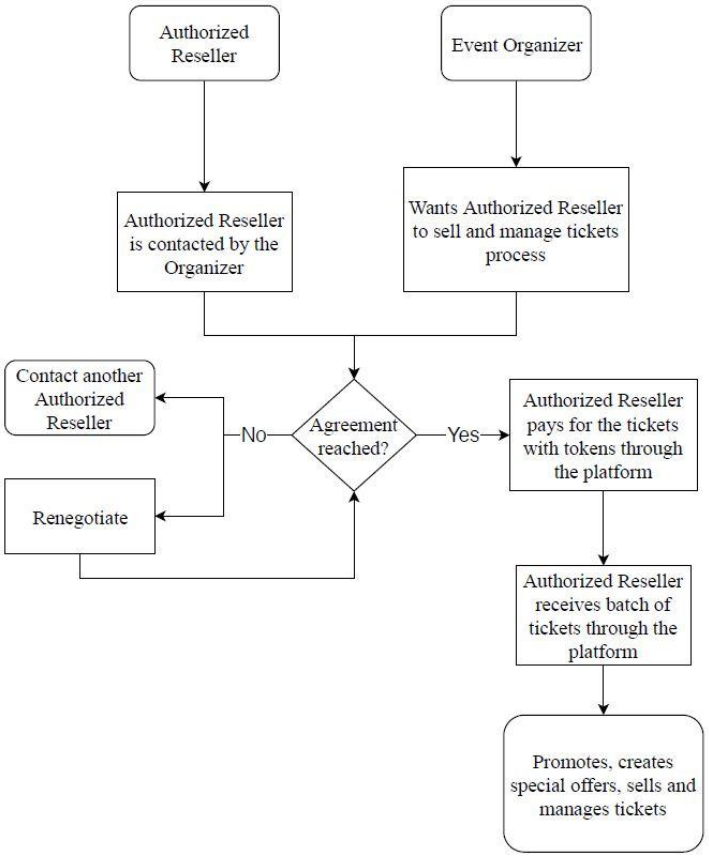

Figure 3.4 - Authorized Reseller Process

Venue Staff: Currently the venue responsible have to verify if the tickets owned by the clients are valid or not. The tickets presented to the venue staff have to be in paper format, printed by the clients or by the authorized reseller that sold the ticket. The proposed solution does not need a ticket in paper format, the ticket will be available for the clients on the platform, through a mobile application developed for that purpose. When the client wants to access the venue, he has to show the ticket to the venue staff, and they will scan the unique identifier of the client's ticket. The scanning of the ticket would be made with a mobile application or with a specific machine for this task. The mobile application would be connected to a secondary database. This database is part of the system and is responsible for the validation and nullification of the tickets during the high peak traffic of the entrance window.

The process flow of redeeming and validation of tickets in the proposed solution is:

1. Ticket sales for the event close hours before the event

2. Tickets are unloaded for a secondary database
3. Venue doors open for the event
4. Venue Staff uses a mobile application to scan client's ticket
5. Request is sent to verify if the ticket is valid
6. Is the ticket valid?
    a. Yes, then the ticket is registered has redeemed
    b. No, the client may not enter the venue

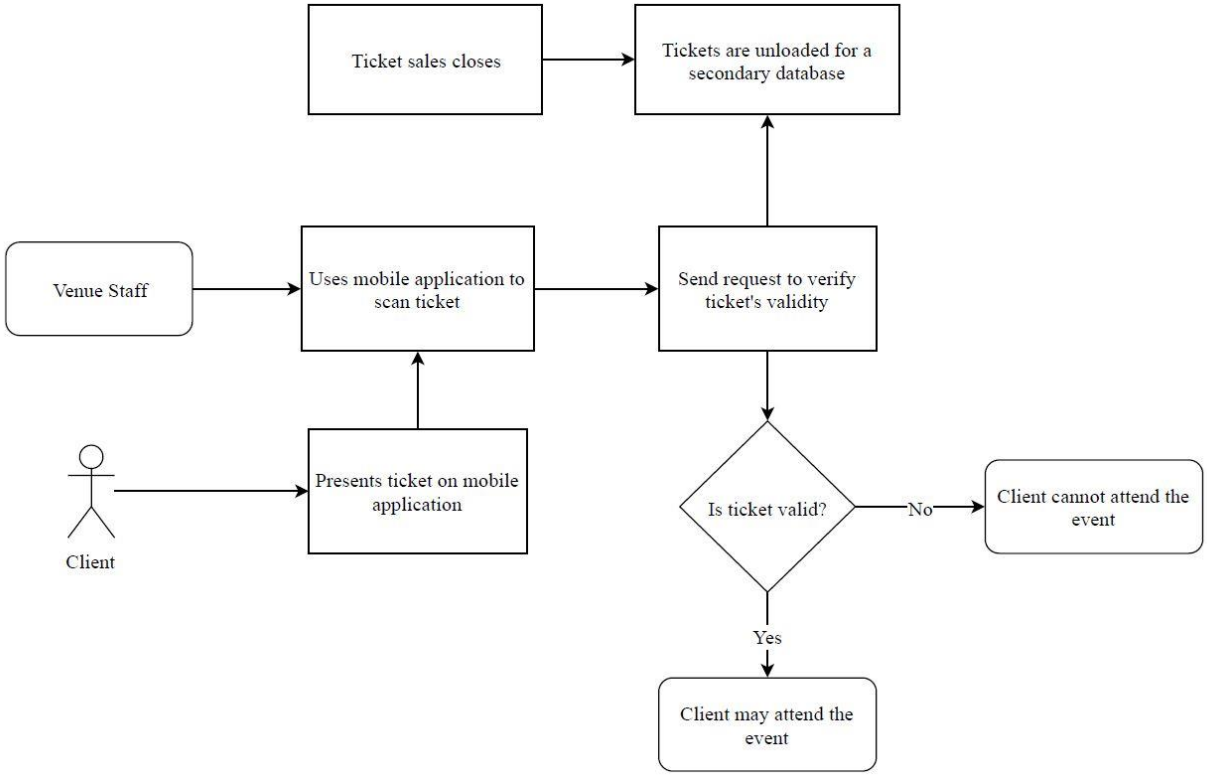

Figure 3.5 - Venue Staff Process

Clients: Within the proposed solution, a client must be registered to access the marketplace and only after being registered with success, he can access and search for events and his tickets. Since the use of cryptocurrencies or tokens was defined as a requirement for this work, for the user to be able to purchase tickets, he must acquire tokens. After acquiring tokens, the user searches the tickets by event and if there are tickets available for the event, he can purchase them. After being bought, the ticket is associated with the user, the user's token amount is updated, and the ticket is available to the user on the platform. With the ticket available to its owner, he can resell it if he cannot attend the event for some reason. The price of the ticket keeps unchanged so, when a person wants to resell a ticket it will be set with the same face value. With this, it is not possible for a regular person to make a profit out of the tickets sold on the platform. Instead of the ticket being sent to the user's email address and then being printed to be validated before entering the event, the ticket is available in the platform in a digital format.

For a client to resell his ticket, the ticket sales for the event must be still open. The ticket sales for each event closes hours before the event, not allowing for any trade between the clients. After the

ticket sales closes, if a ticket set for resell was not sold, the ticket is still owned by the same client who wanted to resell it. This client can attend the event. If the ticket was sold, the client who bought the ticket before the ticket sales closed, is now the new owner of the ticket and the ticket is accessible on the platform. The client who set the ticket for resell, no longer has access to the ticket.

The process flow of purchasing tickets within the solution proposed is:

1. Is client registered on the platform?
    a. No, register client
2. Acquire tokens
3. Searches for tickets for a specific event
4. Are tickets on sale for the wanted event?
    a. No, the sales are still closed or already closed or there are no more tickets for sale for the event
5. Client has enough tokens?
    a. No, acquire tokens
6. Purchases ticket for the event
7. Ticket's ownership is transferred to the client
8. Ticket is available on the web platform and on the mobile application

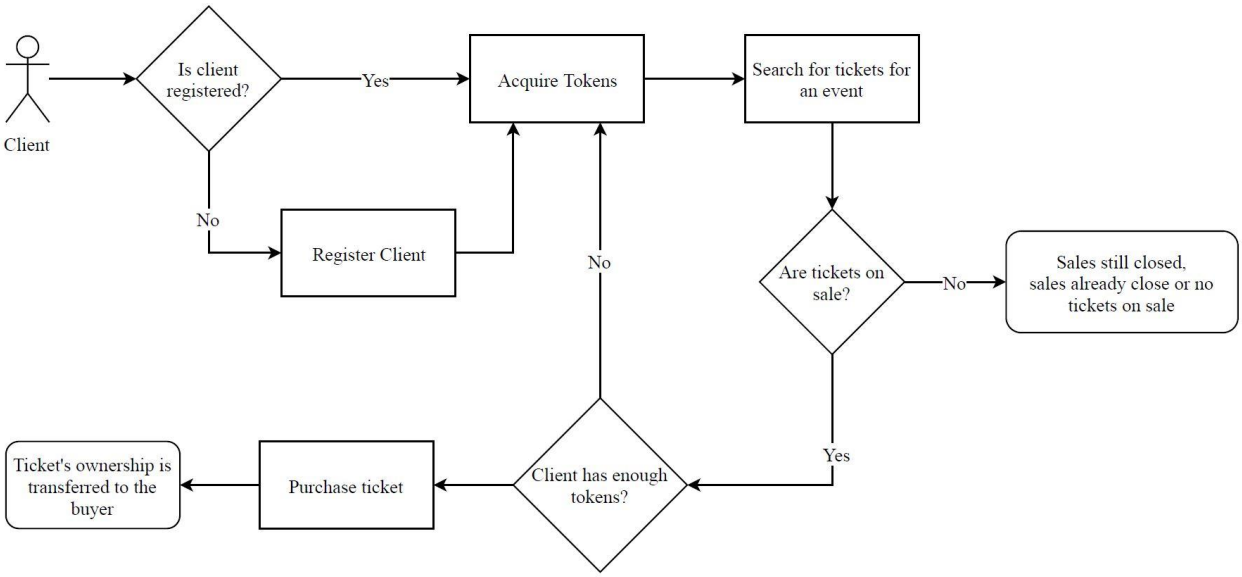

Figure 3.6 - Client Process Flow

The process flow for the resale of tickets by the client in the proposed solution is:

1. Client wants to resell his ticket
2. Access the marketplace
3. Are the ticket sales still open for the event?
    a. Yes, client can put his ticket for sale
    b. No, the ticket cannot be resold
4. Put ticket for sale
5. Ticket is available for other clients to buy
6. Ticket sales for the event closes

7. Was the ticket sold?
    a. Yes, the client does not have access to the ticket anymore
    b. No, the ticket is still owned by the client and he can attend the event

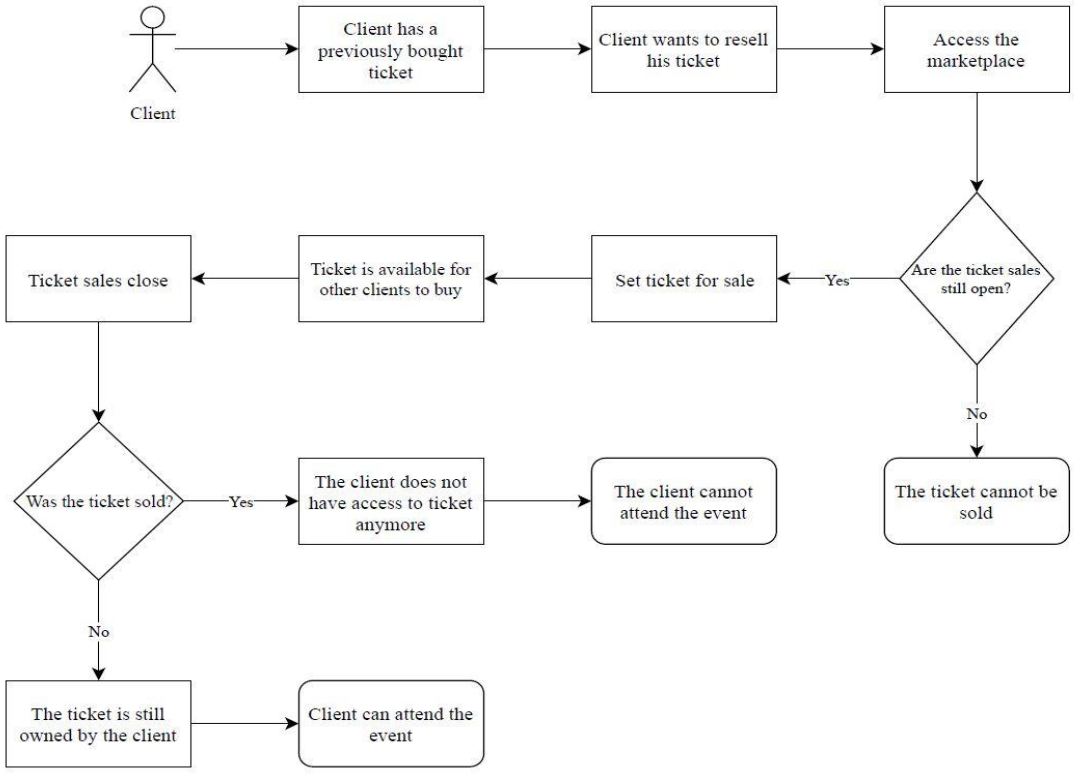

Figure 3.7 - Client Resell Ticket Process

3.2.2 Blockchain's Role

This work addresses the use of blockchain in the ticketing industry, specifically in the process of selling, reselling and redeeming tickets. The proposed solution is a web platform that supports different stakeholders as described before and interacts and registers the transactions related with the tickets on the blockchain. Any transaction that interacts with a ticket and updates its status is recorded in the blockchain and becomes available to be queried later, which allows for a ticket to be tracked at any time. Every type of query is also executed over the blockchain. The blockchain also registers transactions of the events created by the event organizers. The tickets created in the platform are associated with these events.

The transactions registered in the blockchain are made secure using cryptography. Since every block has a unique identification, generated by a hash function, any change made to a ticket generates a completely different hash. The decentralized ledger of data that is used by blockchain, allows for the ledger to be stored on multiple storage devices. This decentralization protects the system from any data loss in case any of the devices that store the ledger face any downtime. This allows for the tickets to be always available, as long as there is one device online registering the transactions. For the ticketing industry, the system availability is an essential requirement that permits the sales not to be impaired by the downtime of the machines.

Chapter 4. Design and Implementation

This chapter addresses the design and implementation of a proof-of-concept of the solution presented in the previous chapter. This work comprised the development of a web platform that works as a marketplace for event organizers, authorized resellers and clients to sell and resell tickets. In this chapter, it is referred as the Smart E-Tickets platform.

4.1 Requirements

This section presents the use cases and non-functional requirements that were considered for the development of the Smart E-Tickets platform.

4.1.1 Use Cases

The Smart E-Tickets platform has 4 types of stakeholders: Event Organizer, Authorized Reseller, Venue Staff and Client. In what follows, we present a use case diagram for each type of stakeholder, representing the different ways these stakeholders can interact with the system, and provide a short description of each use case.

Event organizers, authorized resellers and clients have a wallet to store tokens. The conversion from tokens to fiat currency is executed by a Payment System. These use cases end up not being developed since they were not considered essential for the envisaged proof of concept.

Event Organizer

Create Event

Before an event organizer can sell his tickets, he must create the event on the Smart E-Tickets platform. This operation supports the creation of an event by receiving all the necessary information about the event.

Create Tickets

The operation for creating tickets allows the event organizer to define the tickets' information and create the tickets. These tickets must be associated with an event previously created. Each batch of tickets created by the organizer, will also stay associated with its issuer throughout its lifecycle.

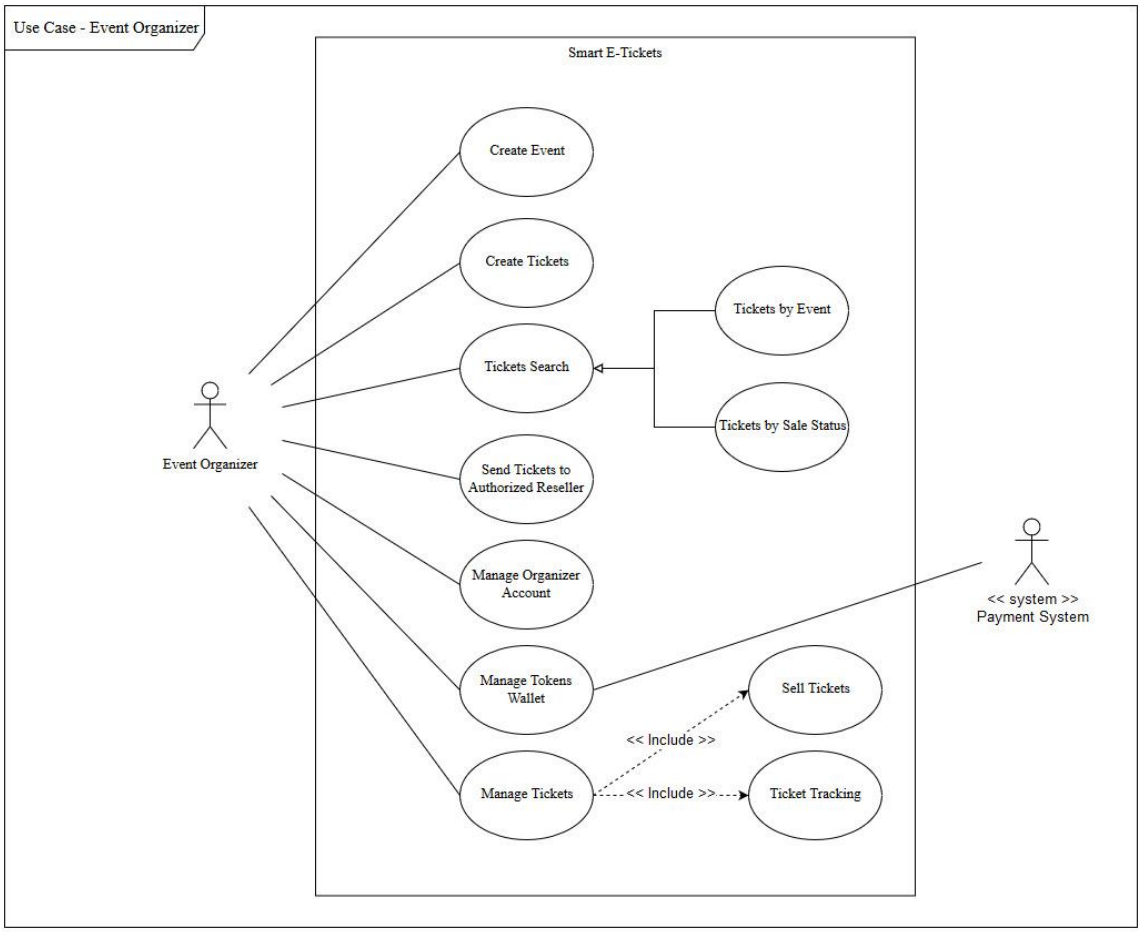

Figure 4.1 – Use Case Diagram – Event Organizer

Tickets Search

An event organizer must be able to search and filter his tickets. This operation allows for a ticket to be searched by the event that it is associated with and by the ticket's current sale status.

Send Tickets to Authorized Reseller

When the event organizer and the authorized reseller reach an agreement for the price of the batch of tickets that the authorized reseller will pay, this operation transfers the ownership of each ticket in the batch. Each ticket has the identification of who issued the ticket and who is the current owner. In this way, the current owner of each ticket is changed to whom the event organizer sells the tickets.

Manage Organizer Account

An event organizer must be able to update his account and any information about the organization that is susceptible to changes. This operation assists the event organizer with this process.

Manage Tokens Wallet

Every user of the Smart E-Tickets platform has a wallet. This operation has several goals. For the event organizer, the goal is to store the tokens that the clients and authorized reseller paid for the tickets.

Manage Tickets

An event organizer must be able to manage the tickets that he issued.

Sell Tickets

An event organizer can sell tickets for his event on the Smart E-Tickets platform. This operation is implicit upon the creation of each ticket and sets the status of the ticket for "For Sale" when it is issued.

Ticket Tracking

This operation allows for the event organizer to track each ticket and have more control and knowledge about the transactions made with them. This tracking is a history of who owned the ticket throughout its lifecycle and any other transaction made with the ticket. Any other attempt of transacting the ticket is recorded along with the account that tried to do it.

Authorized Reseller

Events Search

This operation allows the authorized reseller to search for past and upcoming events and to search events by organizer.

Tickets Search

An authorized reseller must be able to search and filter his tickets and tickets from events that he is not selling. This operation allows for a ticket to be searched by the event that it is associated with and by the ticket's current sale status.

Receive Tickets from Event Organizer

After the event organizer and the authorized reseller reach an agreement for the price of the batch of tickets that the authorized reseller will pay, this operation notifies the authorized reseller that he received the tickets from the event organizer. The tickets become available on the Smart E-Tickets platform for the authorized reseller to manage and sell to the clients.

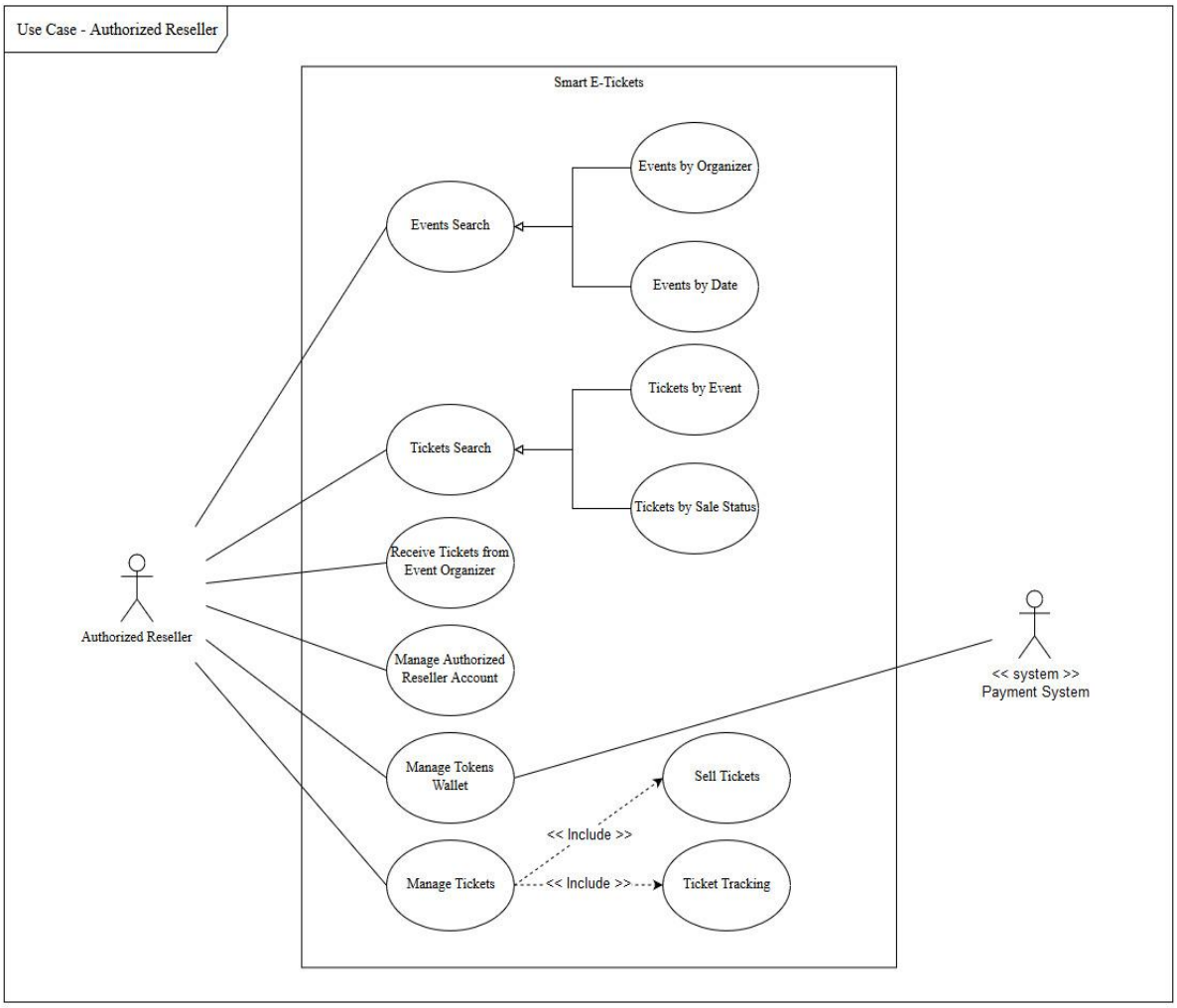

Figure 4.2 – Use Case Diagram - Authorized Reseller

Manage Authorized Reseller Account

An authorized reseller must be able to update his account and any information about the organization that is susceptible to changes. This operation assists the authorized reseller with it.

Manage Tokens Wallet

Every user of the Smart E-Tickets platform has a wallet. This operation has several goals. For the event organizer, the goal is to store the tokens that the clients and authorized reseller paid for the tickets.

Manage Tickets

An authorized reseller must be able to manage the tickets that he bought from the event organizer and create different offers and packages that include these tickets.

Sell Tickets

An authorized reseller can sell tickets that he acquired from the event organizer on the Smart E-Tickets platform. After the authorized reseller receives the tickets from the event organizer, the status of each ticket is set for “For Sale”.

Ticket Tracking

This operation allows for the authorized reseller to track each ticket and have more control and knowledge about the transactions made with them. This tracking is a history of who owned the ticket throughout its lifecycle and any other transaction made with the ticket. Any other attempt of selling the ticket is recorded along with the account that tried to do it.

Venue Staff

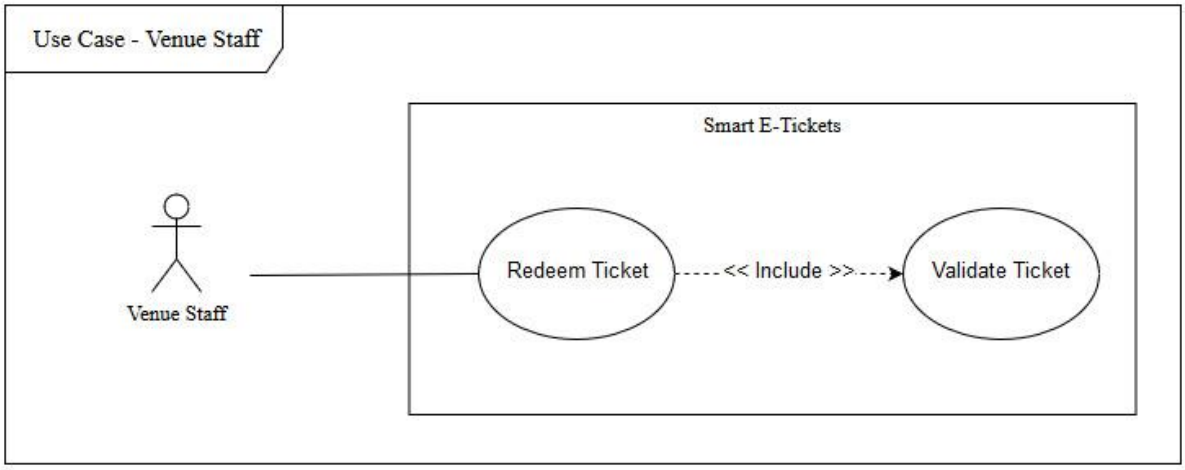

Figure 4.3 – Use Case Diagram - Venue Staff

Redeem Ticket

When a client wants to attend an event, his ticket must be verified and validated, so that he may enter the venue. This operation allows the venue staff to request the validation of a ticket.

Validate Ticket

The validation of a ticket consists in verifying if the ticket was not redeemed already by another client, the impossibility of reselling the ticket after being validated and changing its status to redeemed. The ownership of the ticket cannot change after it’s validation.

Client

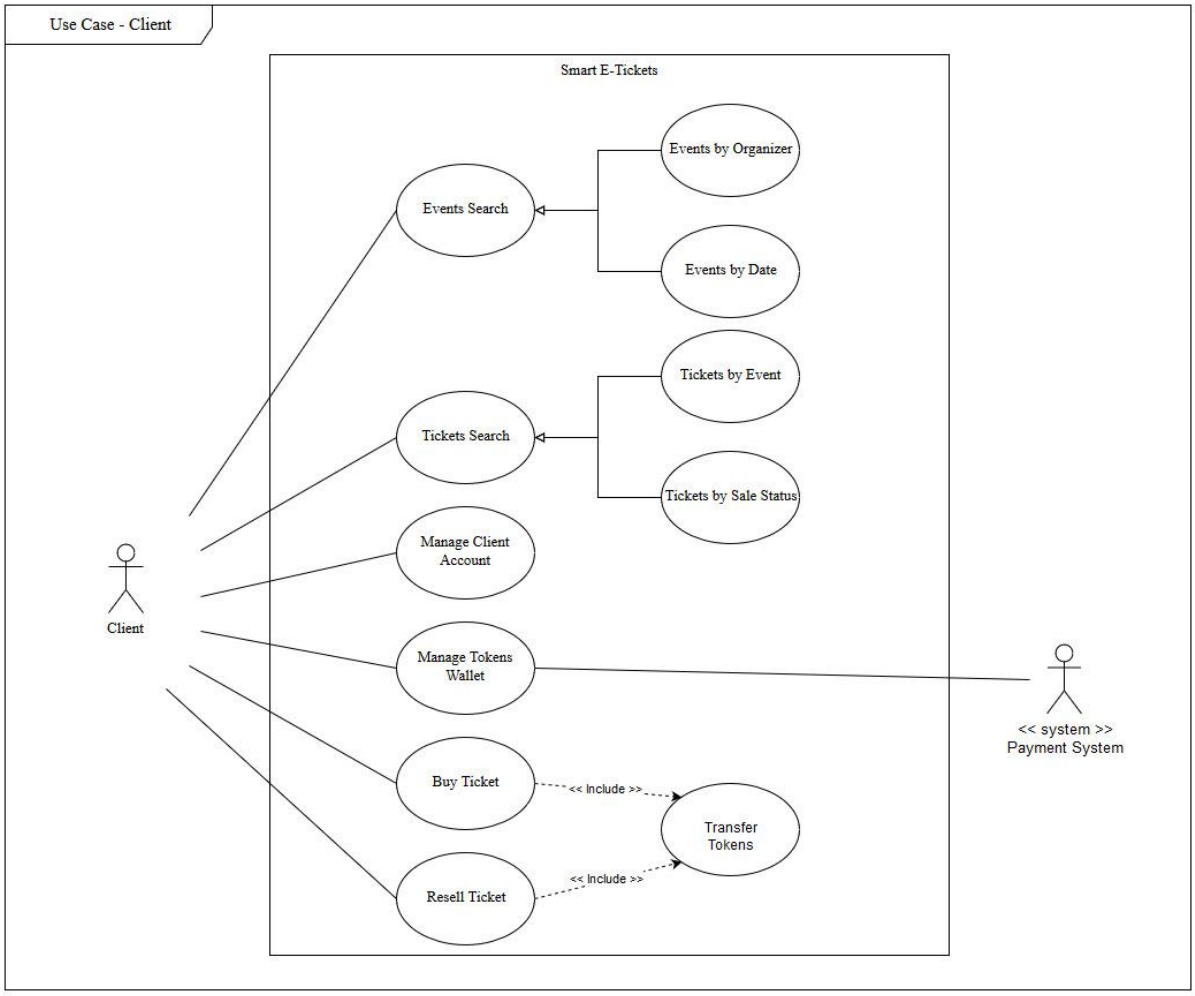

Figure 4.4 – Use Case Diagram – Client

Events Search

This operation allows the client to search for past and upcoming events and to search events by organizer.

Tickets Search

A client must be able to search and filter the tickets the he bought. This operation allows for a ticket to be searched by the event that it is associated with and by the ticket's current sale status.

Manage Client Account

A client must be able to update and delete his account and update any information related to him. This operation assists the client with this.

Manage Tokens Wallet

Every user of the Smart E-Tickets platform has a wallet. This operation has several goals, for the client is to allow him to acquire tokens, so that he can purchase tickets and to store the tokens transacted from the tickets that he resold.

Buy Ticket

This operation allows the client to buy tickets, or directly from the event organizers or from the authorized resellers. The Smart E-Tickets platform only allows for a ticket to be sold by its face value. So, when a client wants to buy a ticket, he needs to have the number of tokens necessary to purchase it. If this condition is met, the ownership of ticket changes from the seller to the buyer and so do the tokens.

Resell Ticket

The operation resell tickets allows the client to set his ticket for sale. Then, the other clients can find his ticket when they are searching for tickets that are for sale. The ticket stays with a status of "Resell" until it is bought by another client or until being redeemed. If is not bought by another client, the client still owns the ticket and can attend the event.

4.1.2 Non-functional Requirements and Constraints

The users of a system have expectations on how well the system will work. The characteristics that are used to quantify the quality of the system include how easy the software is to use, how quickly it executes, its availability during the year and how it handles a high peak of traffic.

For the Smart E-Tickets platform, the following non-functional requirements and restrictions were identified as the most important ones:

1. The system must have a high availability to guarantee that the normal process of ticket sales is not affected;
2. The system must have mechanisms that guarantee the integrity of the tickets, the events and the wallets;
3. The system must have a high performance and availability when the tickets are being validated by the venue staff at the entrance of the venue. With possible multiple entrances working at the same time and with thousands of people to enter the event, the system must be able to perform accordingly to the requests rate;
4. The system must be easy to use by any user, independently of his level of experience;
5. Must be easy to add a new organization such as an event organizer or a reseller to the platform. An organization must be able to start participating at any time;
6. The system must have a well-documented process on how an organization can configure its peers and how to deploy them;
7. Each ticket information must only be accessed by its' issuer and owner;
8. The system must allow an ease of integration with other systems of ticket selling.

The blockchain platform used for the development of the Smart E-Tickets platform was IBM's Hyperledger Fabric. With the adoption of this blockchain platform, the development of the Smart E-Tickets platform was restricted to the platform current capacities.

4.2 Architecture

This section presents the architecture of the Smart E-Tickets platform. As discussed before, the solution encompasses a web application and a blockchain network. An overview of how the peers interact with the applications, the architecture of the peers that form the network and an overview of the whole network used for the development of this work.

4.2.1 Overview

The Smart E-Tickets platform was developed with the goal of facilitating the process of selling and reselling of tickets. Its development had a greater focus on the backend, specifically on the use of a blockchain platform that would replace the traditional database in the system. The platform would host the entire lifecycle of the ticket, meaning that a ticket would never leave the platform and all its trades would take place on the platform. The platform also intends to integrate with the current systems of selling tickets. The idea is not to replace them completely initially but to be a valid option for the clients to trade tickets and for the event organizers and authorized resellers to improve their current way of doing their business. The graphical user interface was developed with the intention of demonstrate how the stakeholders would interact with the platform, focusing on the process of trading tickets.

An overview of the architecture is presented in Figure 4.5. Two different methods of interacting with the system are available: by using a web browser or by using a mobile application. Hence, the system has a web client as well as a mobile client both running in the client side and consuming the REST services provided by the backend. In the backend, besides the web server, there are two data stores and a blockchain network.

For the proof of concept, only the web client was implemented, leaving for future development a mobile application for the clients to access the tickets that they own and their wallets without the need to access a web browser. This mobile application would allow the clients to perform the same actions as they would on the web browser, but being more convenient on the event days to present the ticket to the venue staff to be scanned and validated. For the venue staff, the use of a mobile application or a machine specifically for the job, would allow for them to scan and validate the clients' tickets before entering the venue.

The components that run in the client side allow the communication with the web server by processing and transforming the data provided by the users and routing the requests required trough the services to the web server. A user must be registered and logged in on the platform to be able to access it and execute the basic functions of the platform. The graphical user interface has three types of views and depending on the type of user, there are functionalities shared between them and others exclusive for each type of user. This division allows for a better control of what each type of user can do.

The components in the backend are the most interesting ones and were the main focus of the developed work. The web components, deployed in the web server, have the role of receiving the client requests and initiate the process of communication with the blockchain network. Additionally,

there are other components that allow and assist for the normal functionating of the system: the MongoDB server that assists with the authentication of the users on the platform and stores the users information; the payment system that allows for the conversion between fiat currency and the tokens that are used on the platform for trading tickets; the redeem tickets database, that is used to improve the performance and handle the high request rate of tickets validation during the event days, by dumping to this database the tickets information after the ticket sales closes; and the file system is used to store the identities of the users of the platform.

The blockchain network has restrictions on the number of transactions that can handle per second. This limitation hampers the validation of tickets to be executed directly with the blockchain network because in the event days, there is a high rate of requests expected. This high rate request is due to the multiple doors at a venue working at the same time and that would need to verify a client's ticket by communicating with the blockchain network. Due to this constraint, it was decided to use a secondary database to allow a better performance on the event days.

The blockchain network is composed of three main components: the orderer, that orders the transactions into a block; the peers, that maintains a copy of the ledger and runs smart contracts in order to execute read and write operations to the ledger; and the Certificate Authority, that issues certificates to network member organizations and their users. The ledger stores the tickets, events and wallets created on the platform and the blockchain registers the alterations over time of each of these objects.

The web components communicate with the blockchain network through REST requests or GRPCS requests. GRPCS is a remote procedure protocol that has been mainly used by Google on its cloud products and externally facing APIs [26]. This type of requests is used to communicate with the peer nodes and the orderer, and the REST requests are used to communicate with the certificate authorities of the organizations. An orderer, or an ordering node in Hyperledger Fabric, is a node that does the transaction ordering, combined with other nodes forms an ordering service. The GRPCS was the method of communication adopted by Hyperledger Fabric team for the interaction between the applications and the peers and between peers. It uses protocol buffers for its' serialization layer, that makes the messages exchanged between services small and compact because they are compressed and only contain the data, not the structure of messages. The use of the REST architectural style in the other communications, was due to the broadly adopted use of this type of communication and by the existing know-how of it. By using REST, it is easier to integrate this system with the current systems of ticket sales.

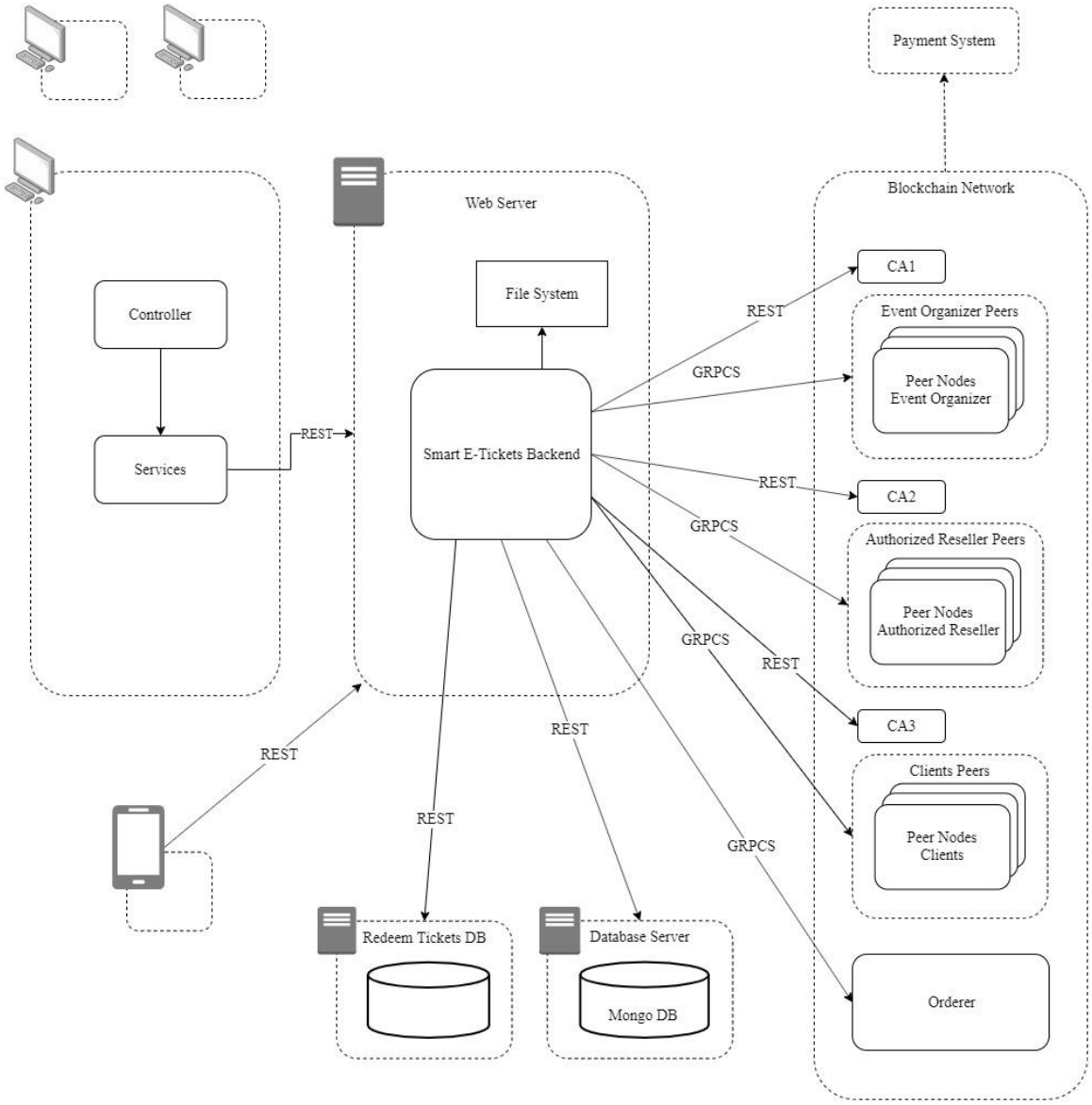

Figure 4.5 – Smart E-Tickets Architecture

4.2.2 Blockchain Network

The Smart E-Tickets blockchain network used in the Smart E-Tickets platform is presented in Figure 4.9. In the next paragraphs, some context for understanding the different elements of the network is provided.

Peer Nodes

Peer nodes are fundamental elements of the blockchain network. They hold copies of the ledger and copies of the smart contracts. Peers expose a set of APIs that enable administrators and applications to communicate with the services they provide.

A peer node can have different structures and is able to host more than one ledger and more than one chaincode. Figure 4.6 shows the configuration of peer nodes used in the blockchain network of Smart E-Tickets platform.

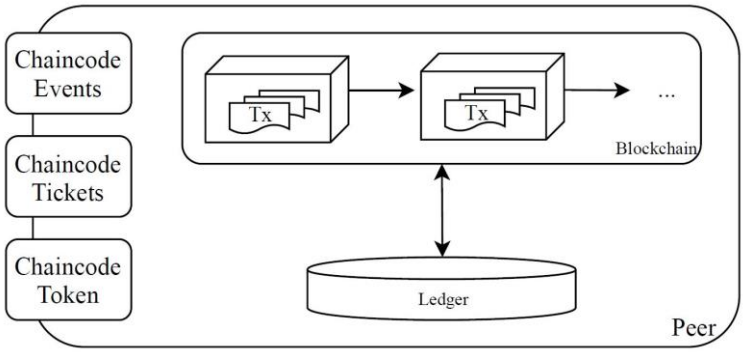

Figure 4.6 – Peer Node Structure, adapted from [27]

Each peer node in the blockchain network holds an instance of the ledger and three smart contracts: a smart contract that has the business logic for the different interactions with the tickets, a smart contract that has the business logic for the interactions with the different events and a smart contract that has the business logic for the wallets. While the blockchain stores the transactions that over different objects, such as tickets, the ledger holds the current state of these objects. For instance, the ledger has the last state of an object after all the changes that affected it, represented as transactions in the blockchain.

Application and Peers

An application needs to connect to the peers to be able to access the ledgers and smart contracts. IBM's Hyperledger Fabric APIs allow the applications to connect to peers, invoke chaincodes, submit transactions and query the distributed ledger. Peers and orderers ensure that the ledger is kept up-do-date on every peer. An orderer, or an ordering node in Hyperledger Fabric, is a node that does the transaction ordering, combined with other nodes forms an ordering service. Any block that a peer validates as generated by the ordering service, is a block that is guaranteed to be final and correct.

Figure 4.7 shows the interaction process between an application and a peer, which comprises:

1. Connect to peer;
2. Invoke chaincode (proposal);
3. Peer invokes chaincode with proposal;
4. Chaincode generates query or update proposal response;
5. Receive proposal response from peer;
6. Request to Orderer that transaction is ordered;
7. Orderer sends transaction to peers in blocks;
8. Peer updates ledger using transactions blocks;
9. Receive update event sent by the ledger.

The Figure 4.7 represents two types of interactions, a query, that goes from the step one to step five, and a transaction, that does all of the steps represented in the figure. The results of a query to an application can be returned immediately, since all the information necessary to satisfy the query is in the peer's local copy of the ledger. A peer does not need to interact with other peers to respond to a query executed from an application. The query process is completed when the application receives the

proposal response. The process of submitting a transaction starts the same way as a query transaction, but this process needs to connect to peers to invoke a chaincode.

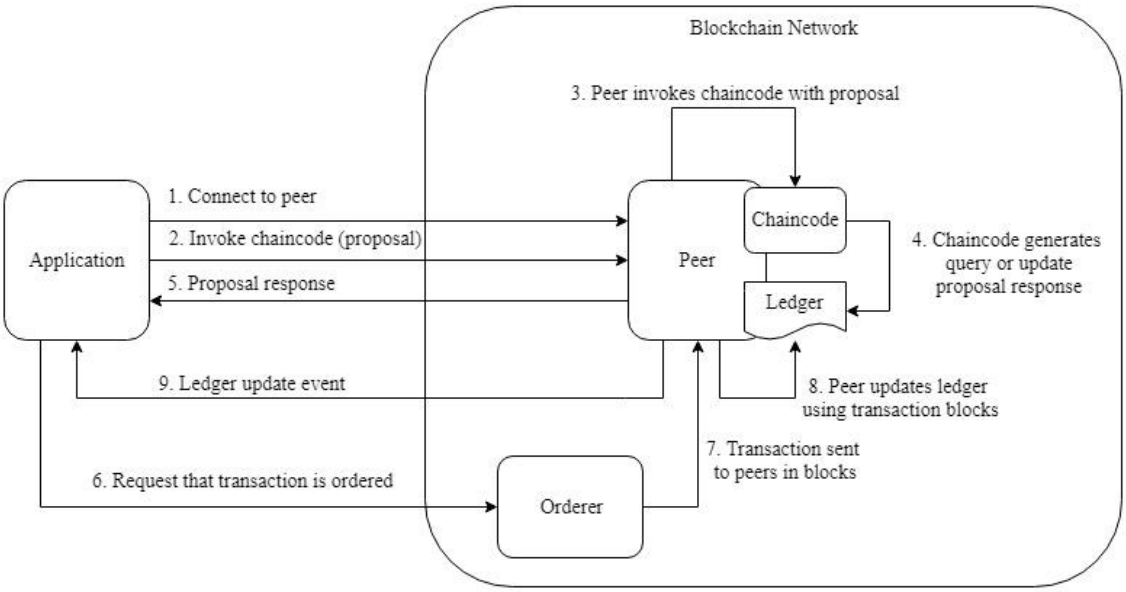

Figure 4.7 – Application and Peers Interaction, adapted from [27]

In this process, a single peer cannot just update the ledger and continue. It is necessary for other peers to first agree to the change, the process called consensus. The proposed update is sent to the other peers by the orderer, who packages the transactions into blocks and distributes them to the entire network. Then, they can be verified before being applied to each peer's local copy of the ledger.

On the Smart E-Tickets platform, the process of querying occurs, for instance, when there is a search for the tickets of an event. The application after connecting to the peers, invokes the smart contract with the parameters that allow for the search of the tickets for the event and the application receives the response with all the tickets for the event. This is where the process of querying is complete. In the case of requesting an update on the ticket, such as the changing of the ticket's ownership, the process continues where the querying process stopped and builds a transaction from all of the responses, which is then sent to the orderer. While for a querying process, it is not necessary to query all the peers, the process of submitting and alteration, all peers must update their local copy of the ledger.

Peers and Channel

Peer nodes, orderer nodes and applications can communicate and transact privately by joining a channel. By joining a channel, they are agreeing to share and manage identical copies of the ledger associated with the channel. The Smart E-Tickets network contains only one channel that is used for all the stakeholders' peers to interact with each other. By only using one channel, the stakeholders have access to the current state of the tickets and wallets, for example, they can see who is the current owner of a ticket and the amount of tokens that a wallet has. The use of only one channel does not guarantee total privacy of tickets data. The stakeholders that are participating in the network, specifically the organizers and authorized resellers, are able to access what is being stored in their local copy of the ledger and consequently are able to access the tickets' information. The clients can only access the tickets that they own, if they resell a ticket, they stop having access to it.

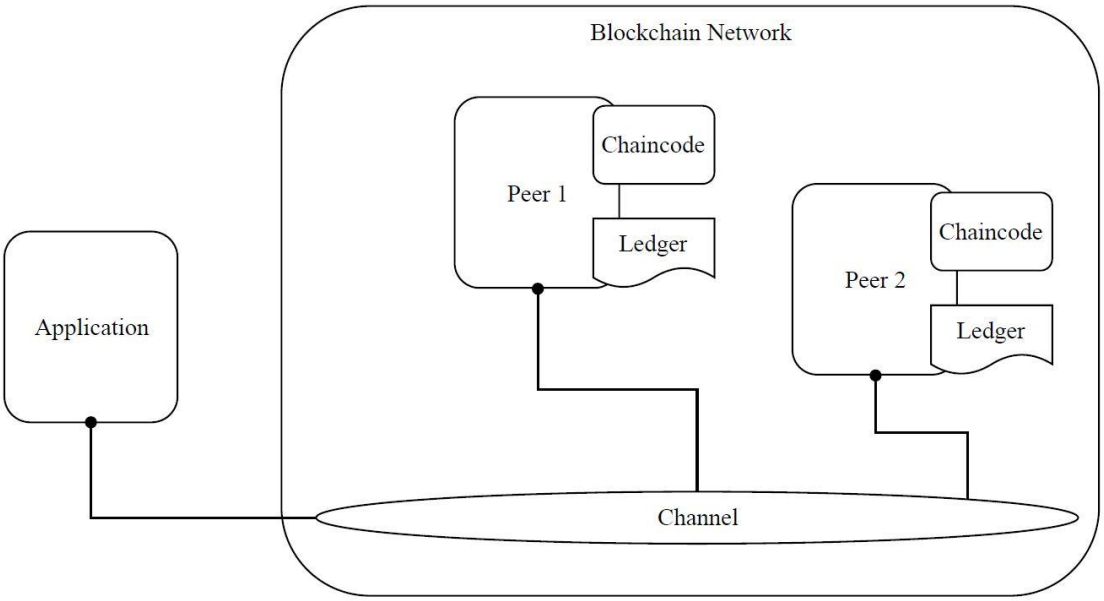

Figure 4.8 – Peers and Channel, adapted from [27]

Peers and Organizations

Private blockchain networks are administered, formed, managed and maintained by a collection of organizations rather than a single organization. Each organization can contribute to the network with as many peers as they want to form the network. The peers owned by each organization do not need to join every channel, but they typically join at least one channel. The organizations participating in the network that was used in this project, are contributing with two peers each and all of them joined the channel available.

Blockchain Network

The Smart E-Tickets blockchain network of Smart E-Tickets platform is presented in Figure 4.9. The network is composed of three organizations: the event organizer, the authorized reseller and the clients. Each organization has a full view of how the network is configured. For each organization there are two peers that contain a copy of the ledger and an instance of each smart contract deployed in the channel. There is an ordering service that services as a network administration point for the network and that supports the channel for the purpose of transaction ordering into blocks for distribution.

The channel configuration and the network configuration are two components that are configured before the deployment of the network. These configurations can be modified after the deployment of the network, but it was not performed in this work. The configurations that were set initially for the network stayed unchanged. The channel used in the network is governed according to the policy rules specified in the channel configuration and the network is governed according to the policy rules specified in the network configuration.

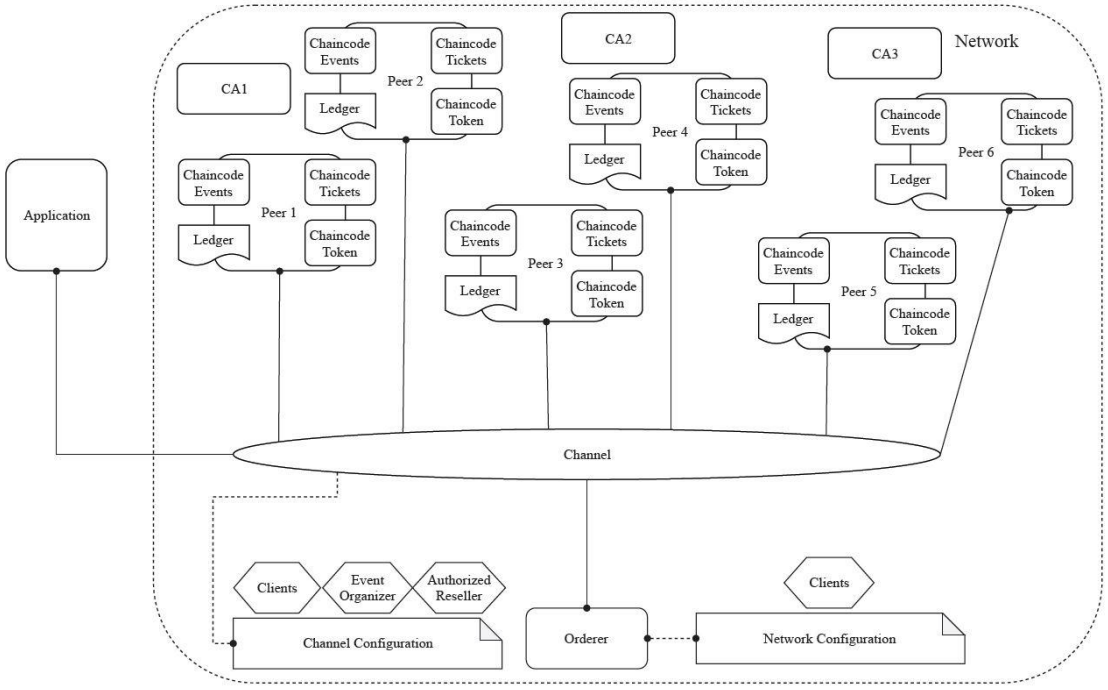

Figure 4.9 – Smart E-Tickets Blockchain Network Overview

4.2.3 Module View

The code of the Smart E-Tickets platform is organized into several layers: presentation layer, application layer, database layer and blockchain layer. The presentation layer is responsible for the user interface of the application and includes two modules, one for the web and another the mobile client. This layer interacts with the application layer, which is responsible for processing the data and translating the REST requests into the internal services. This layer contains four modules: controllers, users service, redeem service and blockchain service. The controllers receive the user's requests and translates them into actions. The users service assists with the registration and authentication of the users, the redeem service would process the request to validate a ticket and the blockchain service gathers all possible interactions with the smart contracts. The business logic layer has the rules for the system, where the most part is inserted in the smart contracts. The redeem business logic was not implemented, but it would validate the tickets in the event days. The users business logic allows for the authentication and registration of the users on the platform and the smart contracts handle the rules for the tickets, events and wallets. These smart contracts are instances that are deployed to the network and where each organization has access to the blockchain network. They also have the role of preparing the data to be inserted into the ledger and registered on the blockchain. The data layer is composed by the redeem tickets, the users and the ledger. The redeem tickets persists the information about the tickets for the event day, the users persists all of the information about the users and the ledger persists the information of the tickets, events and wallets to the blockchain.

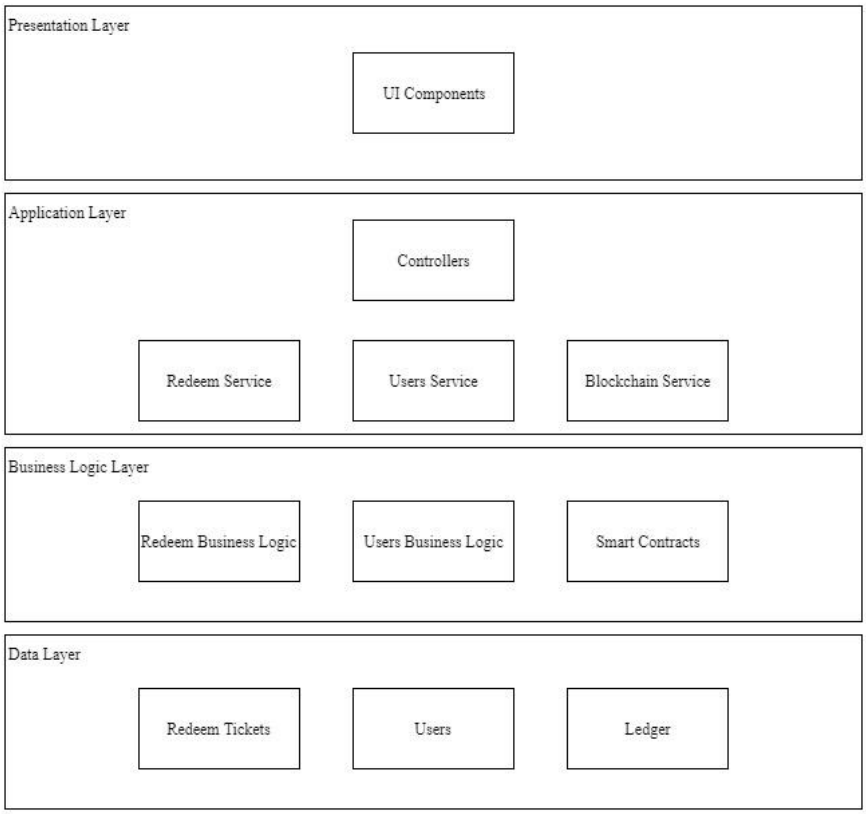

Figure 4.10 - Smart E-Tickets Layers View

4.3 Implementation

This section presents the implementation of the Smart E-Tickets platform, the technologies that were used to setup and deploy the network and the technologies used to develop the user interface, the web server and the smart contracts.

4.3.1 Feature Technologies

The main technologies used for the development of the Smart E-Tickets platform and to deploy the network were the following:

- Angular, a platform and framework that is used for client-side applications in HTML and TypeScript;
- Nodejs, an open source, cross-platform JavaScript run-time environment that executes server-side JavaScript code;
- MongoDB, a database management system that uses a document-oriented database model;
- Hyperledger Fabric v1.4, a platform for distributed ledger solutions with a modular architecture that offers confidentiality, resiliency, flexibility and scalability;
- Docker, a software that performs operating-system-level virtualization and that enables for a fast deployment of the network in containers.

4.3.2 Setup and Deployment of the Network

The setup and deployment of the network was based on the tutorial available at the Hyperledger Fabric documentation, “Build Your First Network” [28]. This tutorial has a sample network consisting of two organizations, each with two peer nodes, and an orderer. This scenario was adapted to the desired network and a couple different changes had to be made due to the lack of dynamism of this tutorial. The tutorial takes advantage of Docker, that allows for instances of peers and smart contracts to be deployed in a virtual machine with little effort. Hyperledger Fabric provides a script that will download and installs samples and binaries. The command used to execute this script was: *curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.0 1.4.0*. This command downloads the version 1.4.0 of Hyperledger Fabric that we chose to use due to its stability. The setup of the network was adapted from a script that is provided by the tutorial that allows for the docker images to quickly bootstrap the network and deploy the network.

One of the options that is given by Hyperledger Fabric at the setup and deployment of the network, is what database it is going to be used in each peer. The state database can be switched from the default, LevelDB, to CouchDB. While LevelDB only supports key and key-range queries, CouchDB allows for rich queries, such as non-key queries. An example of this is that while LevelDB queries can only query the keys of the tickets that are stored, CouchDB allows for queries based on any of the attributes of the tickets, such as, querying by owner, by face value and by issuer of the ticket.

4.3.3 MongoDB Data Structure

MongoDB was used to store information and authenticate the users of the platform. This was achieved with the use of Passport [29], an authentication middleware for Nodejs that supports several strategies of authentication by using a username and password. For this, it was used a passport strategy for authenticating with JSON Web Tokens (as this strategy was found appropriate due to the use of REST in the server) with the following user schema:

- Name: the name of the user that is using the platform;
- Email: the email of the user;
- Username: the username of the user to be displayed on the platform;
- Password: the password of the user that was being stored in the database was a hash of the combination of the user’s password and of a salt. For the generation of the salt and hash was used the bcryptjs library [30];
- Organization: represents the organization of the user that is using the platform;
- Type of User: there are three types of users: the users from an organizer, the users from an authorized reseller and the users that represent the clients;
- Enrolled: represents if it was created a client identity for the user for him to be able to interact with the system that in turn uses the client’s identity to communicate with the blockchain network;
- Wallet Address: represents the identification of the wallet of the user.

4.3.4 Frameworks and SDKs

To be able to connect and interact with the blockchain network, from the server side, it was used the Hyperledger Fabric SDK for Nodejs [31]. This SDK has three features that were used: fabric-network, fabric-client and fabric-ca-client. The fabric-network provides an API for client applications to submit transactions and evaluate queries for a smart contract. The fabric-client provides an API to interact with blockchain network and the fabric-ca-client provides an API to interact with the certificate authorities.

This SDK provides an API that allow us to connect to the blockchain network by using a network gateway. The gateway provides the connection point for an application to interact with the blockchain network. The steps are as follows:

1. Select an identity previously created;
2. Connect to network gateway;
3. Access the network;
4. Construct the request;
5. Submit or evaluate the transaction;
6. Process the response.

Figure 4.11 shows the process of connecting to the network every time a request was made. It needs to read a file that contains the information about the peers that are participating in the network and their URL. Then is setup who is interacting with the network and then the connection is made. With this access, it is then retrieved the network by providing the name of the channel that we want to use. After this, it is just necessary to provide the name of the contract to be able to make the request after constructing it.

```
let connectionProfile = yaml.safeLoad(fs.readFileSync(path.resolve(process.cwd(), 'gateway', 'network_connection.json'), 'utf8'));

// Set connection options; identity and wallet
let connectionOptions = {
    identity: user.email,
    wallet: wallet,
    discovery: { enabled:false, asLocalhost: true }
};
// Connect to gateway using application specified parameters
console.log('Connect to Fabric gateway.');
await this.gateway.connect(connectionProfile, connectionOptions);
// Access Etickets Network
console.log('Use network channel: '+ args.channelName);
const network = await this.gateway.getNetwork(args.channelName);
// Get addressability to eticket contract
console.log(`Use ${this.contractName} smart contract.`);
const contract = await network.getContract(this.contractName, this.contractClass);
```

Figure 4.11 - Connecting to Network Example

Figure 4.12 shows an example of the contract name and class name being used for the contract that would handle the tickets requests.

```
const CONTRACT_NAME = "eticketcontract";
const CLASS_NAME = "org.eticketnet.eticket";
```

Figure 4.12 - Example of a contract and class names

There are two types of requests available: evaluate a transaction and submitting a transaction. When it is just requesting for the system to retrieve from the ledger tickets from an event, it is used the evaluate transaction. The evaluate transaction is evaluated on the peers but the response will not be sent to the orderer and hence will not be committed to the ledger. When an organizer wants to create a ticket, it is used the submit transaction. The submit transaction will be evaluated on the peers and then submitted to the orderer to be committed to the ledger. With both of these functions, it is sent the data to be processed on the peers.

Figure 4.13 and Figure 4.14 show the data used for the creation of the ticket and the use of the function to submit a transaction. Figure 4.15 is an example of how the evaluate transaction request was used.

```
const data = {
  issuer: req.body.eticket.issuer,
  owner: req.body.eticket.owner,
  eventName: req.body.eticket.eventName,
  eventDate: eventDate,
  value: req.body.eticket.value,
  eventID: JSON.stringify(req.body.eticket.eventID),
  eticketsQuantity: req.body.eticket.eticketsQuantity.toString(),
  contract: networkObj.contract
};
let response = await blockchainEticket.createEticket(data);
```

Figure 4.13 - Constructing a Request Example

```
async createEticket(args) {
  const functionName = 'createEticket';
  let response = await args.contract.submitTransaction(
    functionName,
    args.issuer,
    args.owner,
    args.eventName,
    args.eventDate,
    args.value,
    args.eventID,
    args.eticketsQuantity
  );
  return response;
}
```

Figure 4.14 - Submit Transaction Example

```
async getEticketsFromEvent(args) {
  const functionName = 'getEticketsFromEvent';
  let response = await args.contract.evaluateTransaction(
    functionName,
    args.eventName
  );
  return response;
}
```

Figure 4.15 - Evaluate Transaction Example

For the smart contracts, it was used Hyperledger Fabric Nodejs Contract and Shim [32]. The contract provides an interface for the implementation of the business logic of the smart contracts and the shim provides the implementation to support the communication with the peers that form the network for the smart contracts.

4.3.5 Blockchain Data Structure

Three data structures were used and stored in the ledger: the data structure of the tickets, the events and the wallet that holds the tokens. The ledger represents the latest values for all objects ever inserted. These objects are referenced through a key that can be defined as a simple key or as a composite key. There are options for which databases can be used in the peers, the default state database, LevelDB, or an alternative external state database that provides and supports rich queries, CouchDB. In this work was used CouchDB due to its support of rich queries.

The data structure that was used for the events has the following information:

- Key: a composite key of the name of the event, the organizer and the year of the event;
- Name: the name of the event;
- Description: a description of the event:
- Venue: the name of the venue for the event;
- Country: the country where the event will happen;
- Year: the year of when the event will occur;
- Capacity: the maximum number of tickets that can be sold for the event;
- Dates: a list of the dates of the event;
- Organizer: the name of the organizer of the event;
- Type of Event: the type of the event;
- Number of Tickets Created: the number of tickets sold so far;
- Logo: the logo for the event;

The data structure that was used for the tickets is:

- Key: a composite key of the issuer of the ticket, the name of the event for which the ticket is for and the id of the ticket;
- ID: the id of the ticket is a short non-sequential string generated by the shorted library [33];
- Issuer: the name of the issuer of the ticket;
- Issued Date: the issue date of the ticket;
- Owner: the current owner of the ticket;
- Event Name: the name of the event for which the ticket is associated with;
- Event Date: the event date of the event that the ticket is associated with;
- Value: the face value of the ticket that is unalterable;
- Current State: the current state of the ticket, that can be “FORSALE”, “BOUGHT”, “RESELL”, “REDEEMED” and “NULL”;
- Event ID: the id of the event with whom the ticket is associated with.

The data structure that was used for the wallet is:

- Key: a key that is composed by the name of the owner of the wallet;
- Address: the unique identifier of the wallet;
- Owner: the owner of the wallet;

- Token Amount: the amount of tokens that this wallet has;
- Token: the representation of the token that is used. The token has a name and a symbol associated with it.

4.3.6 Smart Contracts

The smart contracts that were created to handle the business logic of the system were: a smart contract for the events, a smart contract for the tickets and a smart contract for the wallet that holds the tokens. The Hyperledger Fabric Nodejs Contract and Shim was the SDK used for the implementation of the smart contracts. The Figure 4.16 represents how the smart contracts are structured. The package lib contains the eticketcontract that works as a controller for the smart contract, it holds all the functions that can be invoked by the applications, that in this cases handles the business logic for the tickets. The eticket represents the object ticket that is going to be stored in the ledger. The ledger-api package is where the requests are built and prepared to be sent to the blockchain and ledger.

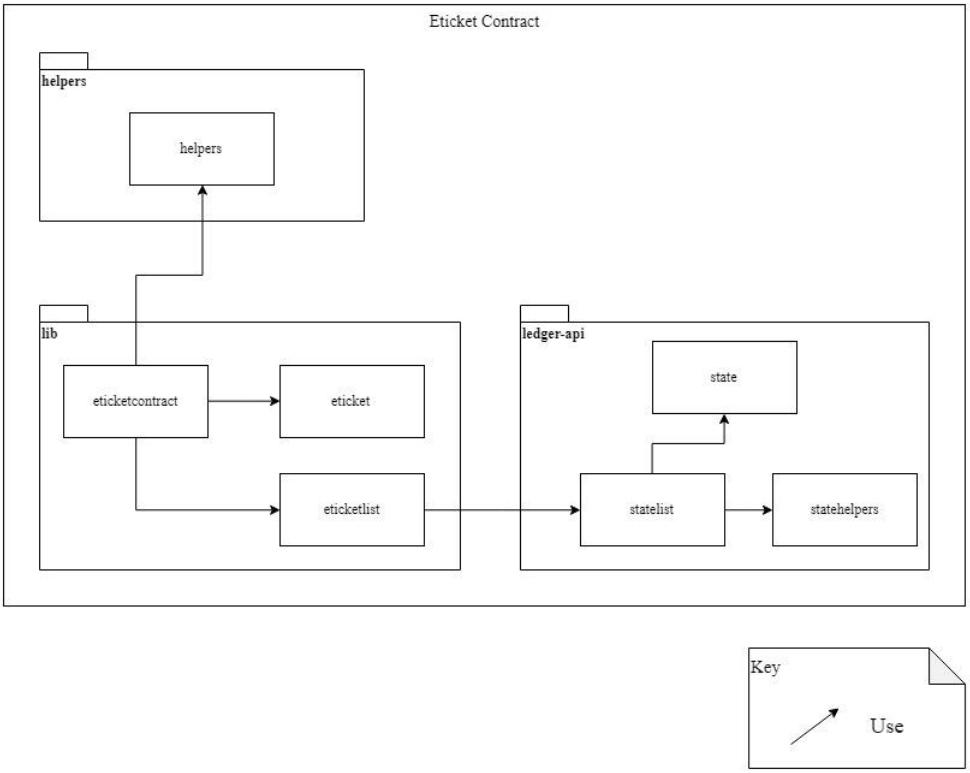

Figure 4.16 - Eticket Contract Structure

Figure 4.17 presents an example of the functions that are used in the eticketcontract. The parameter ctx provides the transaction context per transaction execution. An example is that the ctx allows for the access of the id of who is requesting the current transaction, among other functions.

```
async getEticket(ctx, issuer, eventName, id) {
    const eticket = await ctx.eticketList.getEticket(issuer, eventName, id);
    return Eticket.toBuffer(eticket);
}
```

Figure 4.17 - Smart Contract Get Eticket Function

The Figure 4.18, represents an example of a get request to the ledger. Every object in the ledger and blockchain is represent by a key that is defined when the put request is made. These requests represent the simple queries that the API of the Hyperledger Fabric offers. The Figure 4.19, is an example of a rich query that retrieves from the ledger, events that happened or will happen after a given date.

```
async getState(issuer, eventName, id) {
    let ledgerKey = this.ctx.stub.createCompositeKey(this.name, [
        issuer,
        eventName,
        id
    ]);
    let data = await this.ctx.stub.getState(ledgerKey);
    let state = State.deserialize(data, this.supportedClasses);
    return state;
}
```

Figure 4.18 - Smart Contract Get State Function

```
const queryString = {
    selector: {
        organizer: organizer,
        dates: {
            $elemMatch: {
                $gte: date
            }
        }
    }
};
const stateQueryIterator = await this.ctx.stub.getQueryResult(
    JSON.stringify(queryString)
);
```

Figure 4.19 - Rich Query Example

The smart contracts created for this work followed the patterns that IBM recommended for their architecture. All three of the smart contracts, follow the examples that were show above, each one with its logic but all following the same patterns.

4.3.7 Interface

The user interface of the Smart E-Tickets platform was implemented with the use of Angular Material. Angular Material is a user interface component library for angular that helps build functional web pages and web applications faster by using modern web design principles. For the three types of user of the platform—event organizer, authorized reseller and client—the user interface had the same

appearance, only differing on the actions that each could execute. Figure 4.20 presents an example of the developed user interfaces, showing some of the tickets created by an event organizer. One of the more interesting features that a blockchain offers is the possibility of tracking the changes that a ticket suffered throughout its lifecycle. It was used on the smart contracts a function that returns a history of key values across time. For each historic key update, the historic value and associated transaction id and timestamp are returned. The Figure 4.21, is an example of the transactions that a ticket was involved in. The columns that have more interest in the Figure 4.21 are:

- TxId: this column represents the id of the transaction in the blockchain;
- Timestamp: this column represents the date of when the transaction occurred;
- Owner: this column shows the current owner of the ticket;
- Current State: in this column is represent the current state of the ticket.

The values of these columns, specifically the owner and the current state, are the values that were modified by requests made from a user. The transaction id and the timestamp are to demonstrate that is possible to have an accurate tracking of when the ticket was modified.

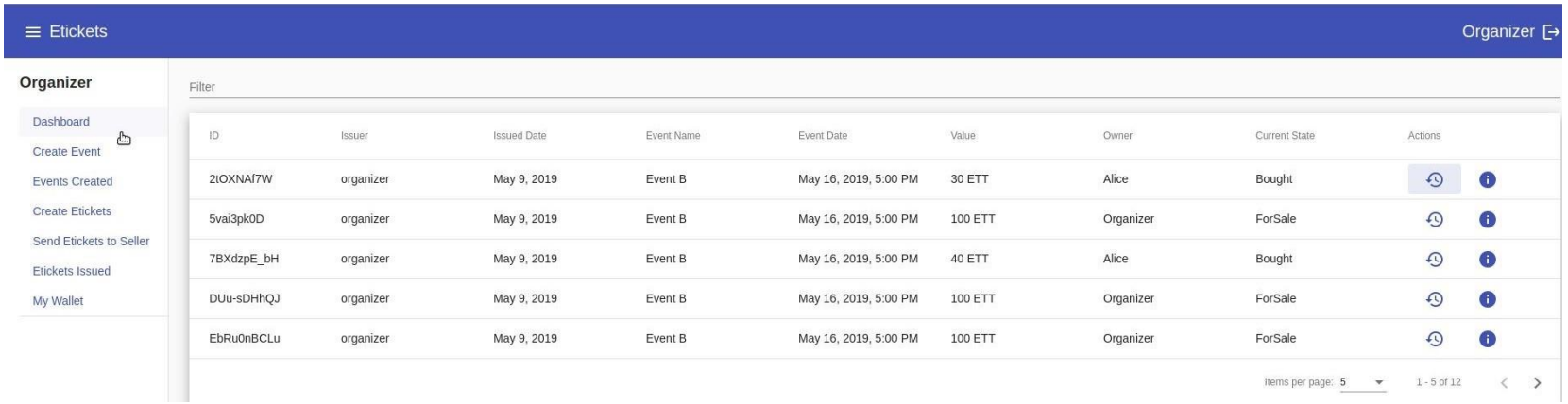

Figure 4.20 - User Interface Example

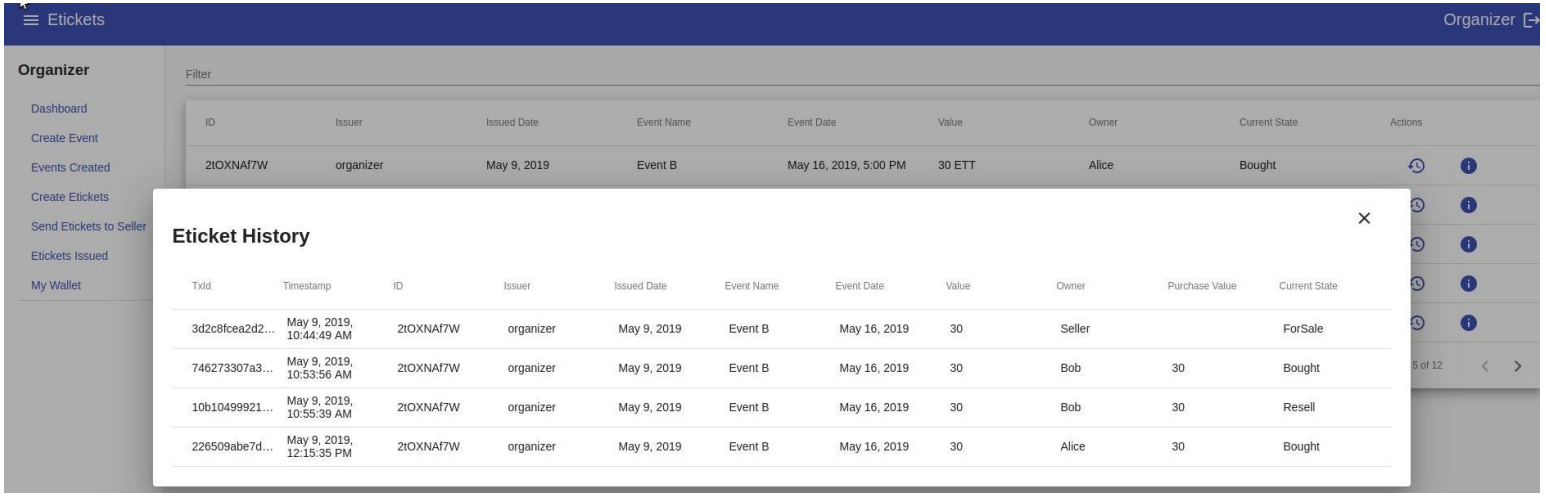

Figure 4.21 - User Interface Ticket History

4.4 Limitations and Tests

The lack of available resources made it impossible to perform the necessary tests to test the system and its features. The tests that would be conducted if they were performed would need to cover the following aspects of the system and of the blockchain platform used:

- Integration tests to cover the different use cases and features of the system;
- Test the blockchain performance when receiving multiple requests per second: test the performance of the blockchain when receiving only query requests, then when receiving requests that have to submit transactions to the ledger and blockchain itself and then test when the system is receiving the two types of requests at the same type;
- Evaluate how the performance of the blockchain it is affected by the number of organizations that participating in the network and the amount of peers that each are contributing to the network;
- Since the data inserted to the blockchain is never deleted, it would be interesting to test how much space it would be necessary to store the tickets information over a period of time, for example, after one year of transactions how much space is the blockchain occupying.

These types of tests would allow for an analysis of how the system and the blockchain would perform under heavy traffic and due to the blockchain data growing over time, how much space would be occupied after a period of time.

Chapter 5. Conclusion and Future Work

This chapter presents an analysis of the work developed in this project and if the technology adds value to the use case and to the ticketing industry. Concluding with, possible future work to add more features, explore the technology even more and improve the system developed.

5.1 Conclusion

During the development of this work at Accenture, I sought to learn about blockchain technology and how it could be beneficial. This was the goal initially proposed by Accenture, to acquire knowledge, so that it could be used in future projects developed by them. The use case of selling and reselling tickets was based on current problems that every person that attends, from small to big events, have to deal with if they want to purchase tickets online. The blockchain technology in this case has the potential to help every stakeholder in these processes to have trust in the tickets that they buy and guarantee that the tickets are not being sold in secondary markets for prices that can go for more than 200% of its face value. The blockchain technology chosen in this project was IBM's Hyperledger Fabric. With the limitations that these technologies still have, with the time that an algorithm of consensus takes and the amount of transactions that can be processed per second, it is necessary for them to be upgraded to allow the technology to be impactful in this use case.

One of the challenges of this work was to setup and deploy the blockchain network. Despite the existence of documentation and tutorials to setup and build the network, the lack dynamism in them, took too much time to be adapted to the network that was used. It was noticed that IBM developed a cloud blockchain platform that allows to build networks faster and easier and operate and govern networks with total control. Concluding that, the ticketing industry can benefit from a technology like blockchain to resolve some of its problems but the platforms that currently exist, need to improve the amount of transactions per second that they support. This would remove the need to use auxiliary databases and interact directly with the blockchain network.

The work developed at Accenture, allowed me to be involved in every step of the selection of a use case to the development of it. This experience taught me some value lessons and the possibility to integrate a company and its daily routines.

5.2 Future Work

With the focus of the work on developing and web platform and a blockchain network, there were some features that were not implemented and processes on the network that were not explored. These future work plans would allow for a more complete system and an added value to the use case.

- Setup and build the network dynamically: with the IBM's cloud platform, it is easier and faster to accomplish this, but it is a service that needs to be paid. Creating an open-source

dynamic solution to setup and build the network, would allow for companies that just want to explore blockchain, to test and configure the network to fit their needs;

- Organizations exiting and entering the network while the network was running: one of the next steps of the project would be to test the entrance and exit of organizations on the network while the network would be running. This would allow to verify how the network would handle the changes in the initial setup;
- Mobile application for the venue staff to validate tickets: one of the next steps of the project would be to develop a mobile application for the venue staff, to allow them to verify the validation of the ticket on the event day;
- Replace the secondary database to validate tickets with the validation being directly made on the network: following the mobile application, the ideal process would be to have it interacting directly with the blockchain network instead of having an auxiliary database.

Bibliography

[1] Instituto Nacional de Estatística, “Estatísticas da Cultura Espetáculos ao vivo : aumento de 43 % nas receitas e 19 % no número de espectadores,” Lisboa, 2017.

[2] Decreto de Lei nº 28/84 de 20 de Janeiro, *Diário da República nº 17/1984, Série I de 1984-01-20*. Lisboa, Portugal: Ministério da Justiça. Ministério da Agricultura Florestas e Alimentação, Ministério do Comércio e Turismo, Ministério da Qualidade de Vida, 1984, pp. 240–258.

[3] Expresso, “Expresso | ASAE detém 24 pessoas por especulação na venda de bilhetes para os U2,” *Expresso*, 2018. [Online]. Available: https://expresso.sapo.pt/sociedade/2018-09-16-ASAE-detem-24-pessoas-por-especulacao-na-venda-de-bilhetes-para-os-U2#gs.ITfTSuw. [Accessed: 28-Nov-2018].

[4] Accenture, “About Accenture,” *Accenture.com*, 2018. [Online]. Available: http://www.accenture.com/us-en/company/Pages/index.aspx. [Accessed: 28-Nov-2018].

[5] Accenture, “Accenture | Innovation Architecture,” *Accenture.com*, 2018. [Online]. Available: https://www.accenture.com/us-en/innovation-architecture. [Accessed: 28-Nov-2018].

[6] P. Curry, “Merkle Trees in 3 Minutes or Less – Coinmonks – Medium,” *Medium*, 2018. [Online]. Available: https://medium.com/coinmonks/merkle-made-palatable-94e6662f4caf. [Accessed: 04-Dec-2018].

[7] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain Technology Overview (NISTIR-8202),” *NISTIR*, p. 59, 2018.

[8] V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper. [Accessed: 05-Dec-2018].

[9] Hyperledger, “Introduction — hyperledger-fabricdocs master documentation.” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/whatis.html. [Accessed: 05-Dec-2018].

[10] Hyperledger, “Blockchain Technology Projects – Hyperledger,” 2018. [Online]. Available: https://www.hyperledger.org/projects. [Accessed: 05-Dec-2018].

[11] Hyperledger, “An Introduction to Hyperledger,” 2018. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf. [Accessed: 05-Dec-2018].

[12] “Corda | Home.” [Online]. Available: https://www.corda.net/. [Accessed: 10-Dec-2018].

[13] D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm (Extended Version).”

[14] “Istanbul byzantine fault tolerant consensus protocol,” 2017. [Online]. Available: https://github.com/ethereum/EIPs/issues/650. [Accessed: 05-Dec-2018].

[15] J.P. Morgan, “Quorum | J.P. Morgan.” [Online]. Available:

https://www.jpmorgan.com/global/Quorum. [Accessed: 05-Dec-2018].
[16] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
[17] W. J. Luther, “Cryptocurrencies, Network Effects, and Switching Costs,” *Contemp. Econ. Policy*, vol. 34, no. 3, pp. 553–571, Jul. 2016.
[18] A. Berentsen and F. Schär, “A Short Introduction to the World of Cryptocurrencies,” 2018.
[19] A. Rai, D. Bhavsar, and Y. Saraswat, “Cryptocurrency: The Emerging or Engulfing Currency,” *Int. J. Innov. Eng. Technol.*
[20] T. Lux and V. Mathys, “FINMA publishes ICO guidelines,” 2018. [Online]. Available: https://www.finma.ch/en/news/2018/02/20180216-mm-ico-wegleitung/. [Accessed: 10-Dec-2018].
[21] GET Foundation Team, “Guaranteed Entrance Token: Smart Event Ticketing Protocol,” 2017. [Online]. Available: https://guts.tickets/files/GET-Whitepaper-GUTS-Tickets-latest.pdf. [Accessed: 10-Dec-2018].
[22] Aventus Protocol Foundation, “A Blockchain-Based Event Ticketing Protocol,” 2018. [Online]. Available: https://aventus.io/doc/whitepaper.pdf. [Accessed: 10-Dec-2018].
[23] “UPGRADED Tickets.” [Online]. Available: https://www.upgraded-inc.com/. [Accessed: 10-Dec-2018].
[24] “Ticketline - Event Tickets.” [Online]. Available: https://ticketline.sapo.pt/en/pagina/faq. [Accessed: 31-May-2019].
[25] “TicketSwap: The safest way to buy and sell e-tickets – TicketSwap.” [Online]. Available: https://www.ticketswap.com/. [Accessed: 31-May-2019].
[26] “gRPC.” [Online]. Available: https://www.grpc.io/faq/. [Accessed: 23-Jul-2019].
[27] “Peers — hyperledger-fabricdocs master documentation.” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html. [Accessed: 17-May-2019].
[28] “Building Your First Network — hyperledger-fabricdocs master documentation.” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/build_network.html. [Accessed: 09-Jan-2019].
[29] “Passport.js.” [Online]. Available: http://www.passportjs.org/. [Accessed: 24-Jul-2019].
[30] “GitHub - dcodeIO/bcrypt.js: Optimized bcrypt in plain JavaScript with zero dependencies.” [Online]. Available: https://github.com/dcodeIO/bcrypt.js. [Accessed: 24-Jul-2019].
[31] “Hyperledger Fabric SDK for node.js Index.” [Online]. Available: https://fabric-sdk-node.github.io/release-1.4/index.html. [Accessed: 24-Jul-2019].
[32] “Hyperledger Fabric Node.js Contract and Shim Index.” [Online]. Available: https://fabric-shim.github.io/release-1.4/index.html. [Accessed: 24-Jul-2019].
[33] “GitHub - dylang/shortid: Short id generator. Url-friendly. Non-predictable. Cluster-compatible.” [Online]. Available: https://github.com/dylang/shortid. [Accessed: 24-Jul-2019].